SONDERHEFT 20 ÖS 100,-/sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 72,- DM 14,-

HAPPY-COMPUTER

2. Atari XL/XE SONDERHEFT

HAPPY-COMPUTER

Markt&Technik

DAS GROSSE HEIMCOMPUTER-MAGAZIN

Atari XL für Kenner

Super Listings

★ Turbo Basic für Kassette
★ 256 Farben gleichzeitig
★ Starke Spiele

Professionell programmieren

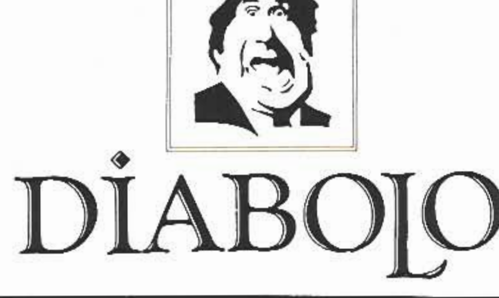
★ Grundlagen Grafik
★ Komplett dokumentierter Registersatz

Tolle Tests

★ Spiele
★ Musik-Compiler
★ Hardware-Freezer

Alle Programme auch auf Diskette erhältlich







★ Der Versand mit den teuflischen Preisen! ★

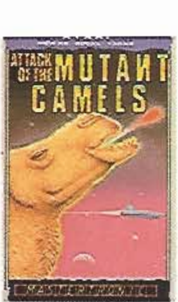
Neue ☎-Nummer: 0 72 52 / 8 66 99



Cass. DM **25.90**
Disk. DM **39.90**



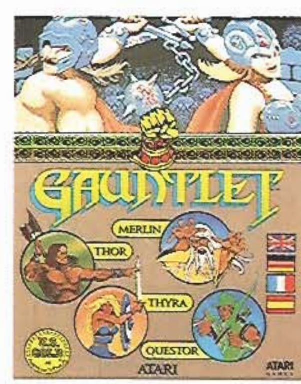
Cass. DM **25.90**
Disk. DM **29.90**



Cass. DM **9.90**



Cass. DM **25.90**
Disk. DM **39.90**



Cass. DM **25.90**
Disk. DM **39.90**

Titel	Cass. / Disk.
Action Biker	9.90 / —.—
Colony	9.90 / —.—
Cristal Rider	9.90 / —.—
Despatch Rider	9.90 / —.—
Gun Law	9.90 / —.—
Hover Bover	9.90 / —.—
Loco	9.90 / —.—
Kik Start	9.90 / —.—
Master Chess	9.90 / —.—
Molecule Man	9.90 / —.—
One Man and his Droid	9.90 / —.—
Vegas Jack Pot	9.90 / —.—
BMX Simulator	14.90 / —.—
Last V.8	14.90 / —.—
LA S.W.A.T.	14.90 / —.—
Ninja	14.90 / —.—
Space Gunner	14.90 / —.—
Spellbound	14.90 / —.—
Red Max	14.90 / —.—
Canon Climber	15.90 / —.—
Designmaster	—.— / 19.80
Shooting Arcade	15.90 / —.—
Actzec	19.90 / 29.90
Boulderdash II	19.90 / 29.90
Cohens Tower/C. Tunnels	19.90 / —.—
Mr. Robot	19.90 / 29.90
Nightrider/Ardy	19.90 / —.—
Spy vs Spy I	19.90 / 29.90
Tale of Beta Lyrae	19.90 / 29.90
Bilbo	—.— / 19.90
Juno First	—.— / 18.90
Max Well's Demon	—.— / 18.90
Mike's Slotmachine	—.— / 19.00
Nibbler	—.— / 18.90
Pyramidos	—.— / 29.00
Tales of Dragons	—.— / 19.00
Soundmachine	—.— / 29.80
Arkanoid	25.90 / 39.90
Space Lobsters	25.90 / 39.90
Sprong	25.90 / 39.90
Starquake	25.90 / —.—
Gauntlet	25.90 / 39.90
Asylum	25.90 / —.—
Boulder Dash	
Construction Kit	25.90 / 39.90
Fighter Pilot	25.90 / 39.90
Ghostbusters	25.90 / 39.90
Goonies	25.90 / 39.90
Hacker	25.90 / 39.90
Hardball	—.— / 39.90

Titel	Cass. / Disk.
International Karate	25.90 / 39.90
Koronis Rift	—.— / 39.90
Leaderboard	25.90 / 39.90
Montezuma's Revenge	25.90 / 39.90
Polar Piere	25.90 / 39.90
Rescue on Fractalus	25.90 / 39.90
Spy vs Spy II	25.90 / 39.90
Super Huey I	25.90 / 39.90
Tomahawk	25.90 / 39.90
Trailblazer	25.90 / 39.90
Spindizzy	25.90 / 37.90
Grid Runner	9.90 / —.—
Stratosphere	9.90 / —.—
Colossum Chess 4.0	25.90 / 37.90
P.O.D.	9.90 / —.—
Micro Rythm	19.90 / —.—
Power Down	9.90 / —.—
Invasion	25.90 / —.—
Astro Droid	25.90 / 29.90
Footballer of the Year	25.90 / 39.90
Mutant Camels	9.90 / —.—
Frenesis	9.90 / —.—
River Ralley	14.90 / —.—
Bubble Trouble	9.90 / —.—

Doppelpack

Cass. + Disk = **19.90** pro Spiel

Clowns + Ballons, Moonshuttle, Pooyan, Sea Bandits, Spider Quake

S★A★M★P★L★E★R★S

★ Greatest Hits vol. 1

Astro Chase · Bristles
Flip + Flop

DM 29.90/34.90

★ 4 great Games

Jet Set Willy · Balloonacy
Pengon · Wizard

Cass. **DM 19.90**

NEU

- Mercenary Kompendium (dtsch.)
  33.90 / 39.90
- The Living Daylights
  —.— / 39.90
- Head over Heels
  25.90 / 39.90
- Auto Duell
  —.— / 49.–
- Ultima IV
  —.— / 49.–

Software-Bestellschein

Ich bestelle aus dem Diabolo-Versand folgende Software:

USA No 352–42–CLN

Anzahl	Titel	Gesamt-preis

Computertyp

Name des Bestellers

Anschrift

PLZ/Ort

Datum/Unterschrift

Ich wünsche folgende Bezahlung:

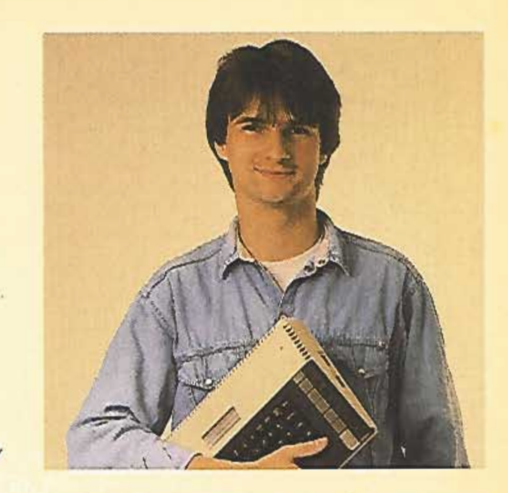
- ☐ Nachnahme (zuzüglich 5.70 DM Versandkosten)
- ☐ Vorauskasse (zuzüglich 3.– DM Versandkosten, ab 100 DM Bestellwert versandkostenfrei)

Bei Vorauskasse bitte Scheck beilegen.

Coupon ausschneiden, auf Postkarte kleben und einsenden an:

Diabolo-Versand, PF 1640, 7518 Bretten.

Eine Abteilung des Verlags Rätz-Eberle GdbR.



Henrik Fisch

Totgesagte leben länger

Sie waren sich alle einig, die Softwarehäuser, die Hardwarehersteller und Entwickler, die Händler, die Fachjournalisten und sogar Atari selbst: »Der Atari XL/XE ist tot«. Schließlich war (und ist) der C64 der bislang erfolgreichste Heimcomputer aller Zeiten. Für Atari schien die einzige Zukunft nach den Wirren der Übernahme durch Jack Tramiel in der neuen ST-Linie zu liegen.

Nur noch einige eingefleischte Freaks und ausdauernde User-Clubs hielten ihrem XL/XE die Stange. Die immer kleiner werdende Gemeinde bombadierte unsere Redaktion mit Leserbriefen: »Bringt doch mehr über den Atari XL«. Keine leichte Aufgabe für uns. Wir bemühten uns dennoch, die Wünsche unserer Leserschaft zu erfüllen. Aus der Listingflut der ersten Jahre war allerdings leider ein immer dünneres Rinnsal geworden. Die Produktion des XL/XE wurde eingestellt, die Händler verkauften ihre Restposten nur noch mühsam. Die Preise fielen. Alles schien klar: Der Atari XL ist tot.

Und plötzlich beginnen die Verkaufszahlen zu steigen. Auf der CeBit '87 kann Alwin Stumpf, deutscher Atari-Chef, stolz 92000 verkaufte XL vermelden. Der Erfolg überraschte Atari derart, daß geplant ist, den 800 XL im Gehäuse eines 130 XE neu aufzulegen. Weiterhin werden von Atari Anstrengungen in Richtung Software unternommen. In diesem Jahr wird eine große Werbekampagne für die »kleinen Ataris« erwartet, ein XL-Computer als Spielconsole kommt zu Weihnachten. Der Atari XL lebt!

Im großen Atari-Schwerpunkt in Ausgabe 7/87 von Happy-Computer konnten es die Freaks aus dem berufenen Munde Alwin Stumpfs hören, daß sie an dieser Entwicklung nicht unbeteiligt waren: »Wenn es nicht die Clique der Atarianer gegeben hätte, die ihren Computer leidenschaftlich vor aller Welt verteidigten, wären die kleinen Computer niemals wieder so populär geworden.«

Bereits das erste XL/XE-Sonderheft von Happy-Computer war ein durchschlagender Erfolg. Das »Turbo-Basic« des späteren GFA-Basic-Programmierers Frank Ostrowski (Listing des Monats in Happy-Computer) ist heute Quasi-Standard aller Listings für die 8-Bit-Ataris. Sogar jenseits des Atlantik. Wer regelmäßig amerikanische Atari-Zeitschriften liest, wird dort immer wieder Programme finden, die in Turbo-Basic geschrieben worden sind.

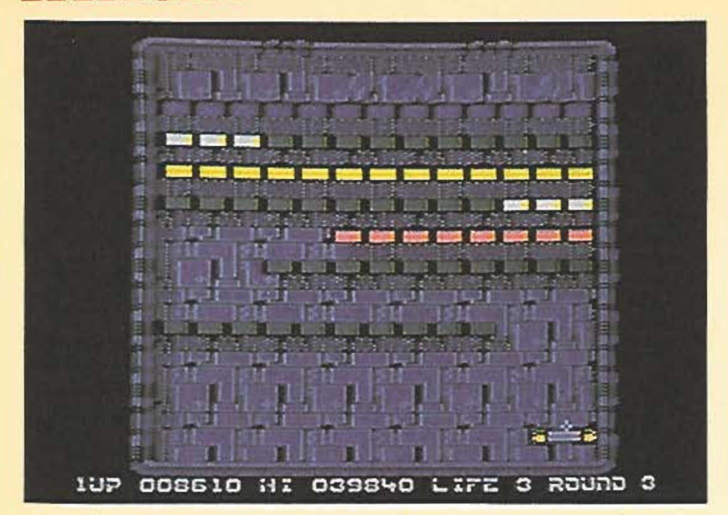
Mit diesem Sonderheft wollen wir dieser Entwicklung noch mehr Durchschlagskraft verleihen. Unter dem Motto »Professionell programmieren« ist dieses Sonderheft ein Nachschlagewerk für alle, die auch noch das Letzte aus ihrem Atari XL herausholen wollen. Wenn Sie die ausführlichste Übersicht über alle XL-Hardware-Register suchen, die je in einer deutschsprachigen Zeitschrift erschienen ist, wenn Sie Grundlagen zur Grafikprogrammierung brauchen, eine Kassettenversion von Turbo-Basic vermissen oder 256 Farben gleichzeitig darstellen wollen: Sie finden es hier.

Für die Anwender unter Ihnen liefern wir unter anderem ein Zeichenprogramm, einige Super-Spiele zum Abtippen und einen Zeichensatzgenerator.

Wenn Sie übrigens ein Spiel oder eine Anwendung programmiert haben, schicken Sie sie uns. Jeden Monat warten in Happy-Computer 3000 Mark auf den Programmierer des »Listings des Monats«.

(Henrik Fisch)

Inhalt



Von jeher wurde der Atari als Spielemaschine eingesetzt. Dieses Talent und seinen Ruf als Spielemaschine hat er sich bis heute bewahrt. Vier neue Spiele für die XL/XE-Computer erwarten Sie ab Seite 6



Nicht nur zum Einfrieren von Programmen, sondern auch als ultimatives Programm-Entwicklungswerkzeug dient der brandneue Turbo-Freezer XL. Einen ausführlichen Test dieses Hardware-Zusatzes finden Sie auf Seite 14



Ein Traum wird wahr: Endlich lassen sich auf dem Bildschirm alle 256 Farben der Atari-Computer gleichzeitig darstellen. Ein Malprogramm und ein Mandelbrot-Programm nutzen die Farbenvielfalt. 21

Spiele

Hardware

Grafik-Listings

Anwendungs-Listings

Spiele-Listings

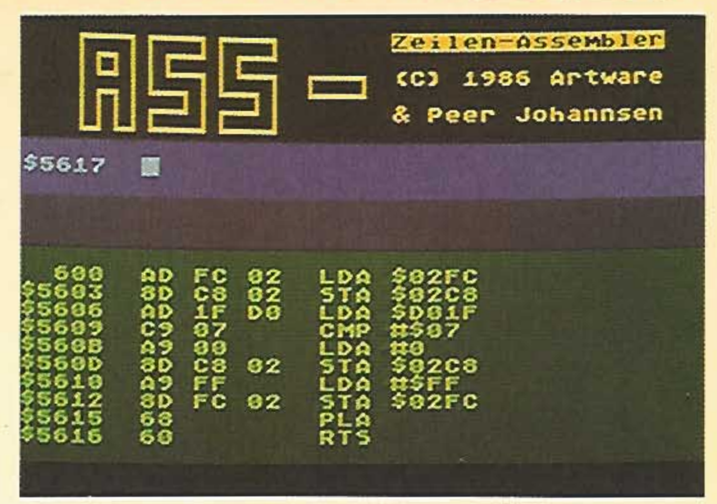
Grundlagen

Software

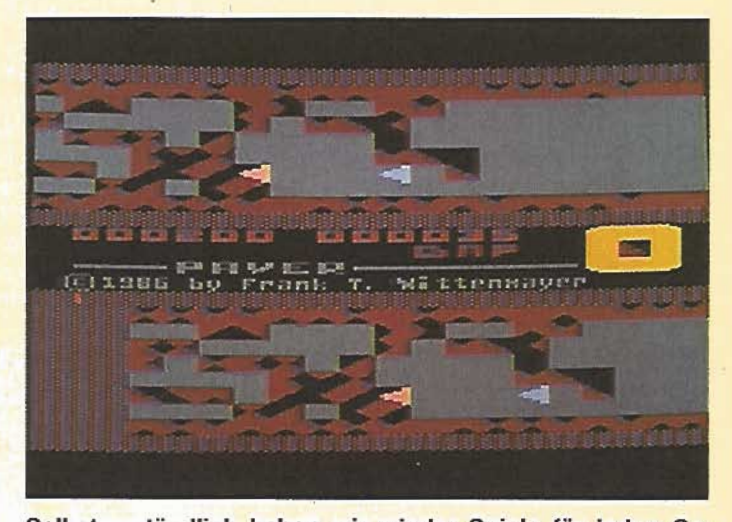
Bastelei

Verschiedenes

● Titelthemen



Mit unserem in Turbo-Basic geschriebenen Zeilen-Assembler lassen sich ohne Probleme kleine Programme schreiben und austesten. Ideal ist er für die Demonstrationsprogramme dieser Ausgabe. **83**



Selbstverständlich haben wir wieder Spiele für jeden Geschmack in diesem Heft abgedruckt. Ob Sie nun lieber Action- oder Denkspiele mögen, hier finden Sie hochkarätige Spiele zum Abtippen. **90**

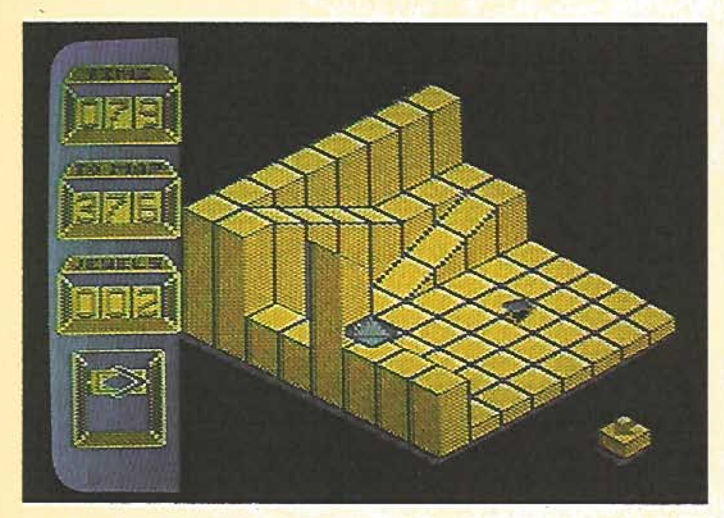


Diese Ausgabe ist als handfestes Nachschlagewerk für die Grafik-Programmierung gedacht. Display-Lists, Zeichensätze und verschiedene Interrupts sind keine Rätsel mehr. Zusätzlich erhalten Sie die Registerdokumentation. **120**



Spindizzy

Eine neue, gut gelungene Umsetzung macht seit kurzer Zeit von sich reden: Spindizzy. Das Spiel wurde vom Schneider CPC so gut umgesetzt, daß jedem XL/XE-Freak das Herz höher schlagen dürfte.



Wie kommt man an den Diamanten ran?

Steile Klippen machen die Erkundung schwierig

Weit entfernt, in einer fremden Dimension haben Wissenschaftler eine bizarre Welt entdeckt, die frei im Raum schwebt. Niemand kann sich erklären, woher sie kommt und wozu sie da ist. Fest steht jedoch, daß sie aus lauter geometrischen Formen besteht. Um diese Welt zu erkunden, wird sofort nach einem waghalsigen Mitarbeiter gesucht, der den Flug zu dieser Dimension und eventuell auftretende Gefahren nicht scheut. Da Sie mal wieder knapp bei Kasse sind, melden Sie sich für den Auftrag.

Sie werden entsprechend ausgerüstet und zum Abflughangar gebracht, wo Ihr Raumschiff auf Sie wartet. Als Sie es erblicken, trifft Sie fast der Schlag. Das Raumschiff ist von so alter Bauart, daß Sie glauben, es müsse jeden Moment auseinanderfallen. Es sieht wie eine auf den Kopf gestellte Pyramide aus und besitzt zudem es ein völlig überholtes Antriebssystem. Dieses läßt das Raumschiff ständig um die eigene Achse rotieren, um so den nötigen Schub für eine Vorwärtsbewegung zu erzeugen (wer sich das wohl wieder ausgedacht hat). Widerwillig steigen Sie in das Fahrzeug und fliegen los.

Als Sie bei der Ebene ankommen, zeigt sich Ihnen ein fantastisches Bild. Wie die Wissenschaftler geschildert haben, besteht alles aus geometrischen Formen. Die quadratische Fläche, auf der Sie sich befinden, hat nach allen Seiten Ausgänge und in der Luft schweben schimmernde Diamanten. Als Sie einen der Diamanten berühren, gibt es einen klirrenden Laut, und der Diamant ist verschwunden. Dafür sehen Sie aber, daß sich Ihr Treibstoffvorrat erhöht hat. Insgesamt können Sie auf diese Weise Treibstoff für 150 Sekunden in Ihrem Fahrzeug lagern. Als Sie sich dem Diamanten nähern, stellen Sie eine weitere Unzulänglichkeit Ihres Gefährts fest: Die Bremsen funktionieren nicht. Um zu bremsen, müssen Sie also entweder genau entgegengesetzt zu Ihrer Fahrtrichtung Schub geben oder Sie benutzen die Notbremse (die <SPACE>-Taste am Computer). Mit dieser kommt Ihr Fahrzeug augenblicklich zum Stillstand. Dafür verbraucht diese jedoch bei jeder Betätigung den Treibstoffvorrat einer Sekunde, und Treibstoff ist kostbar. Ohne ihn bleibt Ihr Fahrzeug stehen, und Sie sind mutterseelenallein auf der Ebene gefangen, ohne je wieder zurückzukehren.

Diese Gedanken schieben Sie jedoch entschlossen beiseite und machen sich auf den Weg. Leicht torkelnd und schwankend fahren Sie eine Rinne entlang und finden sich kurz darauf auf einem Gelände wieder, das zum großen Teil mit Wasser überflutet ist. Ein schmaler Weg führt durch die Fluten zum Festland. Gerade rechtzeitig können Sie noch die Notbremse ziehen, um nicht im Wasser zu versinken. Wie nicht anders zu erwarten, kann Ihr Fahrzeug natürlich auch nicht schwimmen. Weiter führt Sie die Reise durch die fremde Landschaft. Bizarr geformte Klippen wechseln mit unebenem Gelände, endlosen Abgründen und Eiswüsten ab. Dann gibt es wieder Gebiete, die an eine Achterbahn erinnern oder die befürchten lassen, in einem Labyrinth ohne Ausgang gefangen zu sein.

Interessant sind auch die verschiedenen Fahrstühle, mit denen Sie in andere Ebenen getragen werden. Einige dieser Fahrstühle lassen sich nur mit in den Boden eingelassenen Schaltern aktivieren, bei manchen sind gleich mehrere dieser Schalter notwendig. Schließlich entdecken Sie auch, daß Sie nicht allein in dieser Welt sind. Unbekannte Wesen huschen in bestimmten Gegenden umher. Schnell merken Sie jedoch, daß diese Wesen es einzig und allein auf Ihren Treibstoffvorrat abgesehen haben. Mit der Zeit fangen Sie an, sich Gedanken über den Sinn dieser Welt zu machen. Wer hat sie geschaffen und welchen Zweck soll sie erfüllen? Wer weiß, vielleicht lüften Sie das Geheimnis.

Spindizzy wurde zuerst für den Schneider CPC geschrieben, dann auf den C64 konvertiert und ist nun auch in einer Version für die Atari-8-Bit-Computer erhältlich. Wenn auch die Grafik keine so detaillierte Darstellung wie auf dem CPC erlaubt, so ist Spindizzy doch eine rundherum gelungene Umsetzung. Einen großen Vorteil hat jedoch die etwa halb so feine Auflösung der XL/XE-Computer: Die Bildschirme werden wesentlich schneller aufgebaut und die einzelnen Objekte bewegen sich fließender. Spindizzy ist seine 29,90 Mark auf Kassette beziehungsweise 39,90 Mark auf Diskette vollkommen wert. (hf)

Arkanoid

Spielautomaten-Power gibt es auch auf dem Atari! Wer keine müde Mark mehr in die Spielhallen-Automaten investieren will, sollte sich Arkanoid für den XL/XE besorgen.

Die Idee zu dem etwas betagten »Breakout« ist wohl so alt wie die Heimcomputer selbst. Lange Zeit flimmerte das Spielchen in verschiedenen Abwandlungen über die Monitore aller Heimcomputer, bis es nach einiger Zeit in der Versenkung verschwand.

Erst seit kurzer Zeit erlebt die alte Idee ein nie dagewesenes Comeback. Ausgelöst wurde das Breakout-Fieber in der Spielhalle: mit dem Automaten »Arkanoid«. Jetzt ist es endlich auch für die Atari 8-Bit-Computer erschienen.

Die Idee von Arkanoid ist einfach. Sie haben einen Schläger, den Sie am unteren Bildschirmrand hin und her bewegen können. Mit ihm versuchen Sie, einen furios herumsausenden Ball zu treffen. Im oberen Bildschirmdrittel befindet sich eine Mauer, die Sie Stein für Stein abbauen müssen. Jedesmal, wenn der Ball einen Mauerstein erwischt, löst sich dieser (in Wohlgefallen) auf und hinterläßt eine positive Spur auf Ihrem Punktekonto. Wenn Sie alle Mauersteine weggeschossen haben, kommen Sie in einen neuen Level.

Die Hintergrundstory zu Arkanoid ist beachtlich aufgemöbelt, aber eigentlich recht überflüssig für das Spiel: Das Raumschiff Arkanoid wird von einer bösen Macht (von wem sonst...) in einer Raum-Zeitfalle gefangen gehalten. Nur die Raumkapsel »Vaus« hat es geschafft, zu entkommen. Und eben diese Kapsel ist der Schläger, den Sie bedienen. Ihr Ziel ist es, sich durch 32 Screens zu kämpfen, um das Mutterschiff wieder freizubekommen. Damit das richtige Raumschiff-Gefühl aufkommt, ist die Grafik recht abstrakt gezeichnet.

Manchmal sind in den Mauersteinen Zubehörteile verborgen, die man mit dem Schläger einfangen muß. Diese Extras haben die Form von Tonnen, die langsam nach unten fallen. Durch sie kommt Hektik auf. Wenn man nämlich nicht schnell genug ist, kann es passieren, daß man vor lauter Extra-Bonbons den Ball sausen läßt.

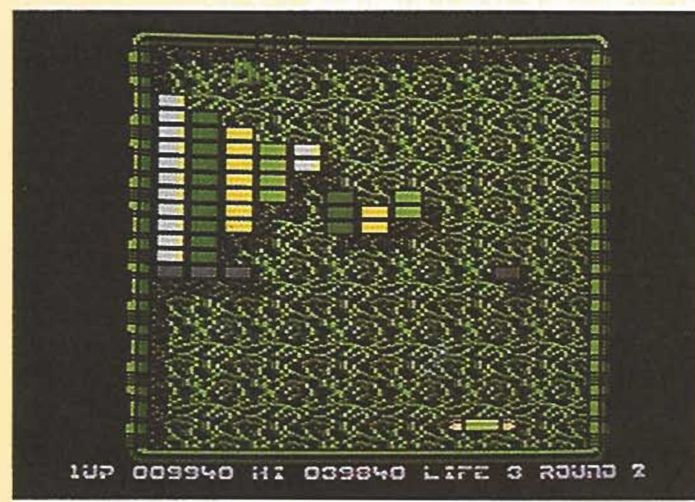
Es gibt insgesamt sieben verschiedene Zusätze, mit denen Sie Ihre Vaus ausstatten können. Ein »P« gewährt ein Extraleben. Wenn Sie ein »G« einfangen, klebt der Ball an dem Schläger und kann so viel präziser wieder abgeschossen werden. Wenn Ihnen zuviel Hektik aufkommt, sollten Sie schauen, daß Sie ein »S« erwischen. Damit wird der Ball langsamer und somit leichter zu treffen. Besonders angenehm ist ein »L«, durch das die Vaus mit einem High-Speed-Laser ausgerüstet wird. Die Mauersteine sind damit in Nullkommanichts abgeräumt.

Das »D« ist etwas für ganz besonders fixe Arkanoid-Spieler. Sobald Sie diese Tonne berühren, verwandelt sich Ihr Ball in drei kleinere Bälle. Natürlich treffen Sie jetzt dreimal so viele Mauersteine, aber Sie müssen auch drei Bälle gleichzeitig im Spiel halten! Mit »E« transformiert sich die Vaus in eine besonders breite Variante. Das wohl begehrteste Extra ist nach wie vor das »B«. Sobald Sie diese Tonne berühren, öffnet sich am rechten unteren Rand die Begrenzung. Wenn Sie mit der Vaus in die entstandene Lücke hineinfahren, kommen Sie automatisch in den nächsten Level. Daß dieses Extra nicht oft in einem Spiel vorkommt, versteht sich wohl von selbst. Oft anzutreffen sind dagegen der Kleber »G« und der »S«low-Modus.

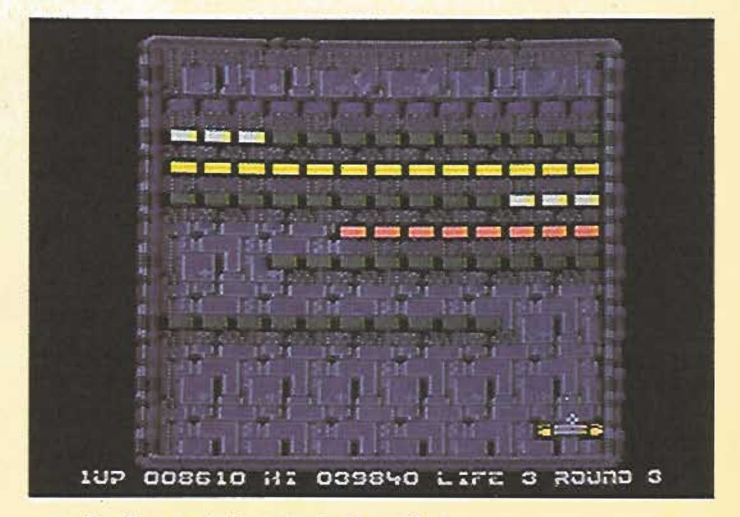
Aber auch mit diesen wichtigen Vergünstigungen werden Sie schwer zu kämpfen haben, denn Arkanoid ist nicht einfach. Spätestens im dritten Level werden Sie Ihre Schwierigkeiten haben. Es tauchen Ziegel auf, die durch nichts zu vernichten sind. Auch ein Laser ist wirkungslos. Sie müssen schon genau zielen, um in den vierten Level zu kommen. Und es stehen Ihnen zweiunddreißig Level bevor ...

Steuern können Sie Ihre Vaus mit so ziemlich allem, was Sie an Ihren Atari anschließen können. Keyboard, Paddles oder Joystick stehen Ihnen zur Verfügung. Die Abfrage ist flott und reagiert gut auf den Spieler.

Grafisch gibt sich Arkanoid auf dem XL/XE etwas zwiespältig. Die Hintergrundgrafik ist absolut dem Automaten nachempfunden und braucht sich nicht vor einem 16-Bitter zu verstecken. Auch die Vaus sieht recht chic auf dem Bildschirm aus. Etwas seltsam ist dagegen der Ball geraten: Vier mal acht Pixel hoch, hat er nicht das beste Aussehen aller Bälle dieser Welt erwischt. Haarsträubend verläuft leider die eckige Flugbahn: Man kann manchmal zusehen, wie der Ball während des Flugs rechnet. Sobald aber die Geschwindigkeit während des Spiels zunimmt, verschwindet der störende Effekt. Insgesamt kann man sagen, daß die Umsetzung für den XL/XE sich nicht hinter anderen verstecken muß. Außerdem ist Arkanoid ein Spiel, von dem man sich nicht mehr so schnell losreißen kann. (al)



Bald haben Sie's geschafft....



...und schon geht's schwierig weiter.



HAPPY-COMPUTER SONDERHEFT

PROGRAMM-SERVICE

Direkt bestellen statt Abtippen

256 Farben gleichzeitig auf dem Bildschirm!

Was Atari in der Werbung versprach und nur bedingt halten konnte, wird mit diesem Programm wahr gemacht: Auf dem Bildschirm sind gleichzeitig alle 256 Farben des Computers darstellbar. Zwei Programme nutzen diese Möglichkeit aus:

Paint256
ist ein Malprogramm, mit dem in 256 Farben gepinselt wird. Neben den üblichen Funktionen zum Zeichnen von Punkten, Linien und Flächen besitzt es eine Funktion zum Einbinden von Text ins Bild und zum Dehnen oder Stauchen von Bildelementen.

Apfel256
berechnet die Mandelbrotmenge in 256 Farben. Ein unvergleichlicher Anblick auf dem Atari Computer!

Arax:
Ein Krieg zwischen den beiden wichtigsten Völkern des Universums ist ausgebrochen. Als junger Kadett starten Sie ein waghalsiges Unternehmen und desertieren von Ihrer Einheit. Sie wollen dem unsinnigen Treiben ein Ende setzen und fliegen zu der Haupbasis des Feindes, um diese zu zerstören. Damit wären dann alle Probleme beseitigt.

Helimann:
In diesem Spiel fliegen Sie mit einem Hubschrauberrucksack augerüstet durch unterirdische Höhlen. Jede Menge Gefahren lauern auf Sie, die Sie bewältigen müssen, um den Ausgang der Höhle zu erreichen. Zusätzlich ist im Spiel ein Editor enthalten, mit dem neue Höhlen entworfen werden.

Paver:
Ein Spiel für zwei. Sie fliegen mit einem Fahrzeug über eine mosaikartige Landschaft, und versuchen, an passenden Stellen Steine einzufügen. Wer als erster die Stelle für den nächsten Stein findet, bekommt die Punkte.

Grafik Draw:
Ein Malprogramm für fast alle Grafikstufen der XL/XE-Computer. Eine Menüsteuerung macht die Bedienung extrem einfach. Fast alle Funktionen werden wahlweise mit einem Joystick oder einem Grafik-Tablet gesteuert. Zusätzlich steht eine Hardcopy-Routine zur Verfügung, die sich einfach an verschiedene Drucker anpassen läßt.

Char Maker:
Das ultimative Zeichensatzprogramm. Alle wichtigen Funktionen werden über den Joystick gesteuert. Unterstützt die Grafikstufen 0, 1 und 12, mit Char Maker lassen sich also auch Farbzeichensätze editieren. Zusätzlich lassen sich die selbsterstellten Zeichensätze in eigene Basic-Programme einbinden.

Weiterhin finden Sie Turbo Basic XL und alle weiteren Programme dieser Ausgabe auf der Diskette.

2 Disketten für den Atari XL/XE
Bestell-Nr.: 25720

DM 34,90* (sFr 29,50/öS 349,–*)

*inkl. Mehrwertsteuer. Unverbindliche Preisempfehlung



Zeitschriften · Bücher
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.



Weitere Programmservice-Angebote für Atari XE/XL

Schnelleres Kopieren auf Atari-Laufwerken

File-Copy: Benötigen Sie ein Programm, mit dem Sie schnell und komfortabel Sicherheitskopien Ihrer Dateien anfertigen können? Wollen Sie Ordnung in Ihre Diskettensammlung bringen? Mit MFCOPY ein Kinderspiel. **Zahlen mit Format:** Mit dieser Routine lassen sich in Turbo-Basic auf einfachste Weise Zahlen runden und formatiert auf den Bildschirm bringen. Diese Funktion ist besonders nützlich für Programme, in denen statistische oder kaufmännische Berechnungen durchgeführt werden. **Tic-Tac-Toe:** Als Endprodukt des Kurses Künstliche Intelligenz selbstgestrickt wurde dieses Programm vorgestellt. Dieses Spiel, bei dem Sie gegen den Computer antreten, veranschaulicht die professionelle Programmierung von Strategiespielen. **Jump:** Sie steuern einen Tennisball, der über eine durchlöcherte Ebene hüpft. Dabei müssen Sie jeden Sprung exakt vorausberechnen und obendrein noch einem Pfeil ausweichen, der sein Unwesen treibt. **The Final Fight:** In diesem fesselnden Action-Spiel müssen Sie Ihr notgelandetes Raumschiff gegen Angriffe verteidigen. Die Bewohner dieses Planeten sehen Ihre Landung als feindliche Kampfhandlung an und versuchen Sie mit allen Mitteln zu vertreiben. Die Anleitung zu diesen Programmen finden Sie in den Ausgaben 4, 5, 6 und 7 der Happy-Computer.
1 Diskette für Atari-XL/XE-Computer

Bestell-Nr.: 20707 **DM 29,90*** sFr 24,90/öS 299,-*

Turbo-Basic – Das Nonplusultra für Programmierer

Turbo-Basic (12/85): Der schnelle Basic-Interpreter für den Atari-Computer. Auf der Diskette befinden sich je eine Version für den Atari 800XL und den Atari 800 mit mindestens 48 KByte RAM. **Ampel (12/85):** Atari-Maschinen-Programm-Eingabe-Listing. **Atari-Prüfsummer:** Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme. **Jumper II:** Listing des Monats 8/84, um die Geschwindigkeit von Turbo-Basic zu demonstrieren. **Magic-Painter:** Listing des Monats 3/85. Ein Zeichenprogramm, das an Turbo-Basic angepaßt wurde.
1 Diskette für Atari 800XL/130XE/800.

Bestell-Nr.: LH8512B **DM 29,90*** sFr 24,90/öS 299,-*

Windows unter Turbo-Basic

Turbo-Windows (9/86): Fenstertechnik für Turbo-Basic. **Drehkörper (9/86):** Ein schnelles Programm in Turbo-Basic zum Zeichnen von 3D-Funktionen. **Labyrinth (3/87):** Unser Listing des Monats März stellt wieder einmal die Krönung der Programmierkunst auf den Atari XE/XL-Computern dar. Sie steuern eine Schlange durch ein Labyrinth und sammeln Gegenstände auf, ohne sich dabei selbst zu beißen. **Morky (1/87):** Schlüpfen Sie in die Rolle unseres kleinen Helden und erforschen Sie insgesamt 35 Räume des Labyrinths, um an Ihr Ziel zu gelangen! **Mord im Computer (4/86):** Als Mark Simpson versuchen Sie, das Verschwinden Ihres Freundes aufzudecken. **Farben gut gemischt (8/86):** Mehr Farben auf dem Bildschirm. **Laufschrift (10/86):** Fehlt Ihnen ein Vorspann für Ihr neues Spiel? Kein Problem mit unserem Maschinenprogramm! **1029 Hardcopy (10/86):** Dieses Programm erledigt schnell und sauber Ausdrucke auf dem Atari 1029-Drucker. **Diashow (11/86):** Zur Präsentation von computergenerierten Bildern. **Nanoconverter (12/86):** Mit diesem Programm können Sie komfortabel und menügesteuert Ihre Spiele von Diskette in den Computer laden. **Drucker-Kosmetik (2/87):** Deutsche Sonderzeichen und Unterlängen sind mit dem Drucker Atari 1029 nicht ohne weiteres darstellbar. Diesem Nachteil wird hier zu Leibe gerückt. **Print-Shop-Grafiken (2/87):** Mit diesem Programm in Turbo-Basic drucken Sie Ihre Grafiken übersichtlich auf Papier.
1 Diskette für Atari-XL/XE-Computer

Bestell-Nr.: 20703 **DM 29,90*** sFr 24,90/öS 299,-*

Grafik unter Turbo-Basic

Turbo-Basic-Interpreter und -Compiler für Atari-Computer • Apfelmännchen in Turbo-Basic • Grafik unter Turbo Basic XL • Daten komprimiert gespeichert • Disksorter • Submission (Labyrinthspiel) • Vorsicht Falle (Action-Spiel) • Pacman mal 2 • Zeichensatzumwandler • Schnellader
2 Disketten für Atari 800XL-/130XE-Computer

Bestell-Nr.: 20703 **DM 29,90*** sFr 24,90/öS 299,-*

Auf der Suche nach dem schwarzen Gold

Ölsuche (8/85): Spiel, bei dem Sie mit dem Atari ST auf Ölsuche gehen. **Atari-Prüfsummer:** Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme. **Geröllheimer (5/85):** Spiel mit Screen-Editor und 20 fertigen Szenen. **24 Farben in Graustufe 0 (6/85):** Routine für farbige Schrift. **Diskhelp (8/85):** Für die schnelle Rettung. **Autostart (9/85):** Basic-Programme starten automatisch. **Dudu 4.0:** Mehr Speicher mit der Floppy 1050.
1 Diskette für Atari 800XL

Bestell-Nr.: LH8510B **DM 29,90*** sFr 24,90/öS 299,-*

Happy-Painter – Das komfortable Malprogramm

Happy-Painter: Unser Listing des Monats der Ausgabe 3/85 ist ein Grafikprogramm, das sich mit anderen Malprogrammen dieser Art messen kann. Besonders gelungen ist die einfache Bedienung, da man mit dem Joystick sowohl im Haupt- als auch in den Untermenüs sämtliche Punkte anwählen kann. Der elektronische Malkasten verfügt über 16 Menüpunkte und bietet eine Grafikauflösung von 160x96 Pixel. **Grafikdemo (3/85):** Alle 256 Farben werden auf dem Bildschirm dargestellt. Eine Farbspielerei, die die hervorragenden Grafikfähigkeiten der Atari-Computer beweist (Rainbow-Effekt). **Variablen-Dump (2/85):** Mit diesem Programm können Sie die verwendeten Variablen eines anderen Programms auf dem Bildschirm listen. **Als die Bilder laufen lernten:** Mit dem Utility Power-Mover können Sie laufende Bilder schnell und problemlos erzeugen. **Statuszeile mit Uhr (1/85):** Damit Sie beim Programmieren nicht die Uhrzeit vergessen, hilft nur eine ständig sichtbare Zeitanzeige. Mit diesem Programm können Sie eine zusätzliche Statuszeile oberhalb des Bildschirms generieren.
1 Diskette für Atari-Computer mit mind. 48 KByte RAM

Bestell-Nr.: LH8503B **DM 29,90*** sFr 24,90/öS 299,-*

Die Jagd nach Gold, Juwelen und Diamanten

Diamantenfieber: Unser Listing des Monats 2/85 ist eine wahre Schatztruhe. Bereichern Sie sich an bunt glitzernden Diamanten, die überall in einem Bergwerk verteilt sind. Aber Vorsicht! Die Stollen sind sehr instabil. Eine falsche Bewegung, und Sie werden von losen Gesteinsbrocken erschlagen. Ein Spiele-Designer sorgt bei unserer exzellenten Boulder-Dash-Variante für anhaltende Spannung. **Die Schatzsuche (Ausgabe 1/85):** Wer möchte sich nicht auch mit einem Schatz bereichern? Wer dazu nicht unbedingt eine Weltreise unternehmen möchte, kann mit seinem Atari 800XL in eine Schatzhöhle eindringen. Gefährliche Tiere wie Skorpione, Ratten und Schlangen erschweren die Suche. **Zeilenzauber (11/84):** Die wichtige RENUMBER-Funktion fehlt leider im Standard-Basic-Editor des Atari. Dieses Programm beseitigt diesen Mangel. **Jumper II (Listing des Monats 8/84):** Ein professionell gemachtes, in Basic geschriebenes Spiel. Exzellente Programmierung, ein High-Score-Zähler und ein eingebautes Demo werden selbst Zweifler schnell überzeugen. Auf musikalische Untermalung wurde großer Wert gelegt, und die verschiedenen Screens sind brillant gemacht. **Mop – Der Goldgräber (7/84):** Schnelligkeit und guter Sound zeichnen dieses Spiel aus. Die Soundfähigkeiten sind wirklich hervorragend. Viele Bilder sorgen für Abwechslung.
1 Diskette für Atari-Computer mit mind. 48 KByte

Bestell-Nr.: LH8502B **DM 29,90*** sFr 24,90/öS 299,-*

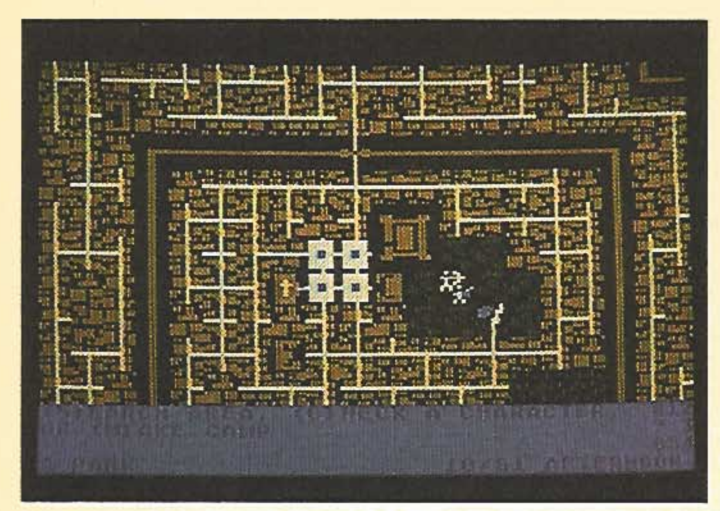
*inkl. Mehrwertsteuer. Unverbindliche Preisempfehlung

Sie suchen packende Spiele, hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.

Bestellungen bitte an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 5656. Österreich: Ueberreuter Media Handels- und Verlagsgesellschaft mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 48 1538-0; Microcomput-ique E. Schiller, Fasangasse 24, A-1030 Wien, Telefon (0222) 785661; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die eingeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Wizard's Crown



Die Kämpfer sind im Stadtpark...



...und müssen sich gegen Riesenspinnen wehren

Ein neues Rollenspiel ist für die kleinen Atari-Computer erschienen: »Wizard's Crown«. Bei diesem Spiel, das ähnlich umfangreich wie »Ultima IV« ist, müssen acht Kämpfer eine Krone suchen.

Es geschah vor 500 Jahren, als der Bund der Zauberer noch über das Land herrschte. Er regierte das Land weise, gerecht und immer zum Vorteil des Volkes. Alle Gerechtigkeit, Güte und Weisheit entstammten einer Krone. Wer diese Krone aufsetzte, besaß alle Weisheit der Welt und konnte Dinge in der Vergangenheit und Zukunft erblicken. Kurz, die Krone war ein absolutes Machtinstrument.

Ein Zauberer war für die Dauer einer vorbestimmten Zeitspanne dazu auserwählt, die Krone zu tragen und das Land zu regieren.

Eines unglücklichen Tages kam es, daß der Zauberer Tarmon an der Reihe war, die Krone an den nächsten im Bunde weiterzuleiten. Der dachte jedoch nicht daran, seine erworbene Machtstellung aufzugeben. So entbrannte ein bitterer Kampf unter den Hexenmeistern, den nur wenige überlebten. Häßliche Kreaturen bevölkern fortan die Ruinen der einst prächtigen Stadt Arghan, von der aus die Magier früher herrschten.

Die letzten Zauberer versuchen verzweifelt, die Krone wieder an sich zu bringen. Kaitar, der älteste der übriggebliebenen, hat dazu einen Plan ersonnen. Er will unerschrockene Abenteurer anheuern, die versuchen sollen, die Krone den Händen Tarmons zu entreißen. Falls sie ihre Aufgabe erfüllen, warten Ruhm und eine so hohe Entlohnung auf die Abenteurer, daß ein Menschenleben nicht ausreicht, das Geld auszugeben.

Soweit zur Vorgeschichte von »Wizard's Crown«, die anderthalb Seiten der 32 Seiten starken Anleitung zum Spiel in Anspruch nimmt. Viel zu lesen also für den Adventure-Begeisterten.

Wizard's Crown ist ein Rollenspiel à la »Dungeons and Dragons«. Soviel wurde in der Vorgeschichte schon verraten: Man spielt mit mehreren Charakteren gleichzeitig. Das ist auch einer der Gründe, warum man sich nicht als Einsteiger an das Spiel heranwagen sollte. Bis zu acht Figuren sollen gleichzeitig kontrolliert werden, wobei die Anleitung nicht ohne triftigen Grund empfiehlt, zu Beginn des Spiels alle acht Charaktere zu nutzen. Das Spiel ist sonst zu schnell vorüber.

Zu Beginn befinden sich alle Charaktere in einer Kneipe, in der sie sich von ihren Streifzügen ausruhen können, erbeutete Schätze und Waffen lagern, sowie im Falle des Ablebens eines Mitglieds einen neuen Abenteurer anheuern können. Von hier aus startet die Kampfgemeinschaft, die Krone zu suchen. Am Anfang sollte man sich jedoch erst einmal auf die Stadt beschränken. Hier gibt es schon genug zu entdecken. Schnell kommt es zu einem ersten Kampf mit einer der zahlreichen Räuberbanden der Stadt. Der Computer stellt dann Kampfmöglichkeiten zur Auswahl: Zum einen gibt es den schnellen Kampf. Hier übernimmt der Computer alle Steuerfunktionen der einzelnen Charaktere und der Angreifer. Zum anderen gibt es den normalen Kampf.

Zu Beginn des Gefechts muß sich die Gruppe erst formieren, um dem Angreifer mit maximaler Kampfkraft gegenüberzutreten. Zauberer sollten sich im Hintergrund halten, und Kämpfer in vorderster Front stehen. Anschließend geht das Hauen und Stechen los. Jeder der acht Gruppenmitglieder kann einen der Angreifer bekämpfen. Dabei spielt die Waffe und Rüstung des Angreifers genau so eine Rolle wie die Position des Gruppenmitglieds zum Feind. Besitzt der Krieger zum Beispiel ein Schild, sollte er sich so hinstellen, daß der Feind links von ihm steht, da er dann einen Schlag des Angreifers mit dem Schild am linken Arm abfangen kann. Insgesamt stehen 21 Funktionen während des Kampfes zur Verfügung.

Haben die Gruppenmitglieder ein Gefecht überstanden, so bekommen sie Erfahrungspunkte, mit denen sie ihre verschiedenen Eigenschaften wie Wendigkeit, Umgang mit der Waffe, Ausdauer und Stärke verbessern. Zauberer brauchen magische Energie, um ihre Kunst betreiben zu können. Nach einem Kampf geht es auch darum, den besiegten Feinden Waffen und Geld abzunehmen. Einige Waffen sind verzaubert und haben eine wesentlich verheerendere Wirkung als die Originalwaffen der Gruppenmitglieder. Waffen, die die Kämpfer nicht gebrauchen können, lassen sich auf dem Markt zu Geld machen. Mit dem Geld wiederum können die Charaktere in einer Kneipe übernachten.

Auch wenn das Spiel extrem kompliziert zu sein scheint (was es übrigens auch ist): Es bringt riesengroßen Spaß. Bis man die Krone gefunden hat und sie Kaitar zurückbringen kann, wird eine ganze Weile vergehen. Und falls man sie wirklich finden sollte, eine Fortsetzung des Adventures ist in Planung. (hf)

Leader Board

Sportspiele gibt es zur Genüge, auch für die kleinen Atari-Computer. Trotzdem hat die Firma Access-Software ihre Golf-Simulation für diesen Computer umgesetzt. Das hat einen guten Grund, denn »Leader Board« ist bislang die am besten gelungene Golfspiel-Simulation, die es zu kaufen gibt.

Bei Leader Board hat man die Wahl zwischen drei Schwierigkeitsstufen, »Novice«, »Amateur« und »Professional«. Bei letzterer spielen Faktoren wie Seitenwind und Anriß des Balls eine tragende Rolle. Diese Stufe sollte man wirklich nur wählen, wenn man schon länger Leader Board gespielt hat und in der Bedienung der vielfältigen Funktionen firm ist. Nach der Wahl des Schwierigkeitsgrades fragt das Programm nach der Anzahl und den Namen der Spieler. Bis zu vier Spieler können gleichzeitig ein Spiel bestreiten, wobei das Programm vier verschiedene Kurse an zu spielenden Löchern zur Auswahl stellt. Kurs 1 ist der leichteste und sollte von dem Anfänger gespielt werden. Bei Kurs 4, dem schwierigsten, sind die einzelnen Bahnen so mit Wasserlöchern und Inseln gespickt, daß es arge Schwierigkeiten bereitet und hoher Genauigkeit bei der Dosierung des Schlages bedarf, um eine Bahn zu meistern.

Das Spiel beginnt damit, daß ein kleines Männchen, zusammen mit der Ansicht der Bahn, aus der Perspektive des Männchens gezeichnet wird. Jetzt legt man mit dem Joystick mittels eines Kreuzes auf dem Bildschirm die Schlagrichtung fest. Ein kleiner Pfahl seitlich des Spielers informiert über die herrschende Windrichtung und die Windstärke. Der Schlag muß nun in dieser Richtung ausgeführt und so dosiert werden, daß der Wind kompensiert wird. Anfangs keine leichte Aufgabe, denn die Kontrollen bei Leader Board sind genauso kompliziert wie funktionsgerecht.

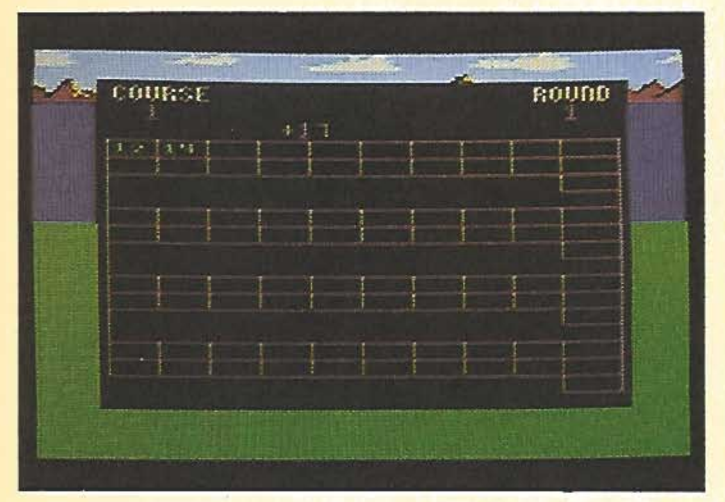
Durch den ersten Druck auf den Joystickknopf legt man die Schlagstärke, durch den zweiten Druck den Anriß des Balls fest. Je nach gewähltem Schläger fliegt der Ball nun eine bestimmte maximale Strecke. Die maximalen Schlagweiten der einzelnen Schläger sind in dem Handbuch aufgeführt. Da der Computer ständig die Entfernung des Spielers zum Loch anzeigt, hat man mit aufgeschlagenem Handbuch neben dem Computer schnell heraus, welchen Schläger man wählen muß. Ging der Schlag nun völlig daneben oder fiel der Ball ins Wasser, so gibt es einen weiteren Versuch von genau der gleichen Position aus.

Ist der Ball dagegen auf dem Festland liegengeblieben, so wird ein neuer Bildschirm gezeichnet. Falls der Ball nun innerhalb eines Radius von 64 Fuß zum Loch gelangt sein sollte, hat man nur noch einen Schläger zur Verfügung. Dieser »Putter« genannte Schläger ist dazu da, den Ball sachte ins Loch zu spielen. Je nach der Anzahl der Schläge, die man für eine Bahn benötigte, werden vom Programm Punkte vergeben. Für jede Bahn ist dabei eine bestimmte Anzahl Schläge vorgegeben. Braucht man weniger Schläge, werden Punkte dazugezählt, bei mehr benötigten Schlägen werden Punkte abgezogen. Es kann also durchaus vorkommen, daß man am Schluß der 18 Bahnen eines Spiels eine negative Punktezahl hat.

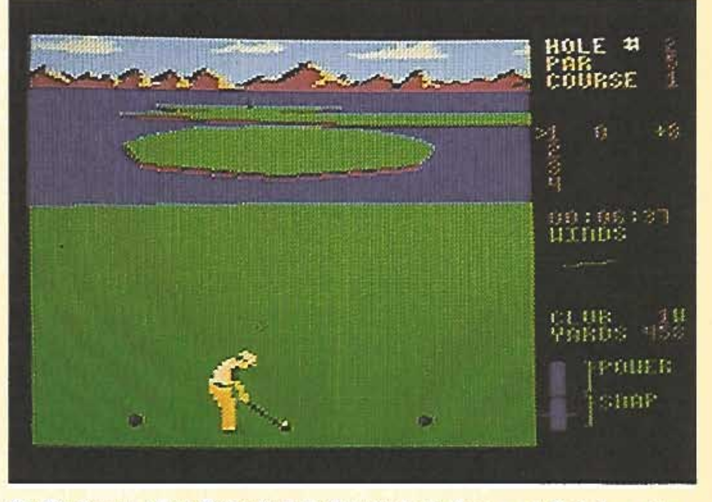
Beim Auspacken der Diskette fällt ein kleiner Stecker auf, der am Joystickport 2 aufgesteckt wird. Dieser Stecker, im Fachjargon »Dongel« genannt, dient als Raubkopierschutz des ansonsten ungeschützten Programms. Das Programm fragt periodisch diesen Stecker ab. Sollte er einmal während des Spiels abgezogen werden, so wird der Bildschirm sofort mit wüsten Linien überzogen, das Programm verweigert in dem Fall die Arbeit. Bei der Diskette befindet sich eine 16seitige Anleitung, in der die besondere Bedienung des Spiels beschrieben wird. Auch wenn den Anfänger die Bedienung des Spiels etwas kompliziert anmutet – hat man das Prinzip der Funktionen erst einmal begriffen, ist die Bedienung kinderleicht.

Wie wohl alle Sportspiele bringt Leader Board mit mehreren Spielern wesentlich mehr Spaß als allein. Da die Simulation durch gute Programmierung einen Bezug zur Realität herstellt, kann man durchaus Geschmack auf richtiges Golfen bekommen. Die Grafik und die Kontrolle über die Schläge ist sehr gut. Leider fehlen wie in anderen Versionen Bäume in den Bahnen. Trotzdem ist Leader Board ein Spiel, das immer wieder gerne hervorgeholt werden wird.

Leader Board ist auf Kassette und Diskette erhältlich und kostet 39 Mark in der Kassetten- sowie 59 Mark in der Diskettenversion. (hf)



Mit der Anzeigetafel hat man immer den Überblick



Golfbahnen aller Schwierigkeitsgrade laden zum Spielen ein

XL tiefgekühlt



Mit einem Freezer lassen sich Sicherheitskopien geschützter Programme anfertigen. Der »Turbo-Freezer XL« bietet zusätzlich ein DOS, einen Monitor mit Disassembler, 256 KByte RAM und das Betriebssystem der alten Atari Computer.

Was ist ein »Freezer«? Ein Freezer (englisch 'to freeze', einfrieren) ist ein Computerzusatz, gleich ob Hardware oder Software, der ein beliebiges Programm im Computer anhält und auf Diskette oder Kassette speichert. Anschließend läßt sich das Programm wieder laden und an genau der gleichen Stelle fortsetzen, an der es »eingefroren« wurde. Auf diese Weise läßt sich von jedem Programm, das nicht während des Laufens Teile nachlädt, eine Sicherheitskopie anfertigen, auch wenn der Kopierschutz noch so gut ausgeklügelt ist.

Für den C64 gibt es Freezer schon länger. Die Atarianer mußten auf diesen nützlichen Zusatz jedoch bislang verzichten, da die interne Struktur der Atari 8-Bit-Computer es sehr kompliziert macht, Programme an einer beliebigen Stelle anzuhalten und ohne Veränderungen wieder zum Laufen zu bringen. Bernhard Engl, der den Atarianern durch sein Turbo-1050-Modul für die Diskettenstation bereits ein Begriff ist, hat nun einen Freezer für den XL entwickelt.

Der Turbo-Freezer XL entpuppt sich als ein 8,5 mal 10,5 Zentimeter großes mit elektronischen Bauteilen vollgepacktes Platinchen, das hinten auf den Erweiterungsbus des XL aufgesteckt wird. An der Seite der Platine steht ein kleines Kabel ab, das die Stromversorgung des Freezers sicherstellt. Ans Ende des Kabels ist eine schmale Hülse angelötet. In der Anleitung wird darauf hingewiesen, daß der Turbo-Freezer XL ganz ohne Lötarbeit zu installieren ist. Deshalb wird die 5-Volt-Versorgungsspannung mit dem Kabel am Joystickport abgegriffen. Die Hülse wird zu diesem Zweck auf den unteren zweiten Pin von links an einem der beiden Joystickports aufgesteckt. Am Computer montiert, sieht der Freezer dann ein wenig abenteuerlich aus, denn die offene Platine steht ein gutes Stück über den Rand des Computergehäuses hinaus, und das Kabel am Joystickport macht auch keinen sehr vertrauenerweckenden Eindruck. Da hätte man sich doch ein stabiles Gehäuse für die Platine gewünscht. Mit Blick auf den Preis des Freezers darf man jedoch nicht meckern.

Wenn man den Computer einschaltet, merkt man zunächst überhaupt nichts vom Freezer. Wir waren im Basic-Modus, und der Computer meldete sich ganz normal mit »READY«. Auf der Freezer-Platine befindet sich ein auffälliger kleiner roter Knopf. Wenn man diesen drückt, erwacht der Freezer zum Leben und übernimmt das Kommando über den Computer. Der Freezer hat jetzt schon sämtliche Prozessor-Register und Hardware-Register gespeichert und die Werte in seinem RAM verwahrt. Man befindet sich dann im Hauptmenü des Freezers. Dieses besteht aus sechs Funktionen, die durch einfaches Tastendrücken aufgerufen werden. Zunächst gibt es die Funktion »Save extern«, mit der der gesamte Computerspeicher auf Diskette oder Kassette gespeichert wird. Bei Diskettenbetrieb läßt sich zusätzlich bestimmen, ob ein vom DOS lesbares File erzeugt werden soll, oder einfach nur die Diskettensektoren beschrieben werden. Um nicht allzuviel Platz zu verschwenden und die Lade- und Speicherzeiten in Grenzen zu halten, wird der Computerspeicher in einer kompakten Form abgelegt. Dieser Datenblock enthält auch sämtliche von Freezer zum Zeitpunkt der Aktivierung gesicherte Registerwerte.

Sicherheit geht vor

Mit »Exec extern« wird ein so gespeicherter Datenblock wieder in den Computer geladen, sämtliche Register restauriert und das Programm an genau der gleichen Stelle fortgesetzt, an der es unterbrochen wurde. Wir haben einige Programme mit dem Freezer ausprobiert und konnten keins finden, das nach dem Auftauen nicht lief, oder das nach dem Starten irgendwelche Fehler zeigte. Mit dem Freezer hat man also die Möglichkeit, von kopiergeschützten Programmen Sicherheitskopien zu ziehen. Interessant ist auch, ein Spiel an einer gefährlichen Stelle einzufrieren und zu speichern. Verliert man sein Spielerleben, kann man mit dem Freezer sofort an der gespeicherten Stelle weitermachen und muß sich nicht erst wieder durch sämtliche Gefahren kämpfen.

Mit dem dritten Menüpunkt, »Zero OS RAM«, wird das RAM unter dem Betriebssystem gelöscht. Programme, die diesen Speicher nicht benutzen, werden dadurch in ge-



2/87
Markt&Technik
68000er
DAS MAGAZIN DER NEUEN COMPUTER-GENERATION
FEBRUAR 1987
DM 7,–
Atari ST
• Wichtige Kaufhilfen für Einsteiger
• Vorgestellt: die besten Programme
• Riesen Hardware-Überblick
• Multitasking mit GfA-Basic
• Macintosh-Emulatoren im Test
Amiga
Grafik-Sensation Deluxe Paint II
QL
Große Assembler-Parade
Mac
kreativ!
Interview mit »Axel F.« – Komponist Harold Faltermeyer
Viele anspruchsvolle Listings für Atari ST, Amiga und QL
AUFBRUCH IN EINE NEUE DIMENSION
mit »68000er«, dem Magazin der neuen Computer-Generation
▶ Programmiersprachekurse für Basic, C, Modula und Assembler.
▶ Bauanleitungen für professionelle Hardware-Erweiterungen.
▶ Spiele-Spaß und -Spannung auf höchstem Niveau.
Ihre hot-line zur Spitzentechnologie von AtariST, Amiga, Macintosh und Sinclair QL.
Das »68000er«-Magazin erscheint
jeden Monat neu!
POSTER & GUTSCHEIN
KOSTENLOS FÜR SIE
84 mal 60 Zentimeter High-Tech-Szene erwarten Sie! Ihr »68000er«-Poster ist im Abonnementpreis enthalten und gehört Ihnen, auch wenn Sie Ihre Bestellung widerrufen sollten.
FÜR EIN KOSTENLOSES PROBEEXEMPLAR DES »68000er«-MAGAZINS
JA, ich möchte »68000er«, das Magazin der neuen Computer-Generation, kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »68000er« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte es dann regelmäßig frei Haus per Post. Außerdem nutze ich den Abonnement-Preisvorteil von 8% und bezahle pro Jahr nur 77,– DM statt 84,– DM im Einzelverkauf.
Vorname
Name
Straße
PLZ, Ort
Datum 1. Unterschrift
Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.
Datum 2. Unterschrift
Gutschein ausfüllen und absenden an: Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar
HCS20



ATARI

dBMAN

Das relationale Datenbanksystem für Ihren Atari ST

dBMAN ist ein leistungsstarkes und flexibles Datenbanksystem

Ihre Daten werden damit relational aufgebaut und verarbeitet. Doch das ist noch nicht alles: Mit der integrierten, bedienerfreundlichen Programmiersprache von dBMAN entwickeln Sie Ihre individuellen Anwendungsprogramme. Sie erstellen Auftragsbearbeitungsprogramme und andere komplizierte Anwendungen. dBMAN ist zu dBASE III kompatibel.
Als Umsteiger haben Sie dadurch die Möglichkeit, sofort mit dBMAN zu arbeiten.

Leistungsmerkmale:

- O dBMAN erzeugt vollständige Datenbanken. Sie können auch im nachhinein ohne Datenverluste die Struktur verändern und erweitern.
- O Sie verwalten Ihre Daten mit dem mitgelieferten ASSIST(enten), ohne zeitraubend ein Handbuch wälzen zu müssen.
- O Das Zufügen, Löschen, Verändern, Darstellen und Ausdrucken von Daten erfolgt auf einfache Weise.
- O Sie können bis zu 10 Dateien gleichzeitig öffnen, 2 Billionen Datensätze pro Datei verwalten und eine unbegrenzte Zahl von Variablen verarbeiten.



Lieferumfang:

- O umfangreiches deutsches Handbuch
- O 3½"-Diskette
- O Beispieldiskette

Hardware-Anforderung:

Atari ST mit mindestens 400 Kbyte freiem Speicher (1Mbyte RAM oder 0,5 Mbyte und TOS im ROM), SM-124- oder SC-124-Monitor, mindestens 1 Diskettenlaufwerk.

Bestell-Nr. 51109
DM 399,–* (sFr 345,–/öS 3990,–*)

* inkl. MwSt. Unverbindliche Preisempfehlung

Diese Markt&Technik-Softwareprodukte erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder im Buchhandel. Wenn Sie direkt beim Verlag bestellen wollen: gegen Vorauskasse durch Verrechnungsscheck oder mit der abgedruckten Zahlkarte.

	Version	Best.-Nr.	Format	Preis DM	sFr	öS
dBMAN (Vers. 3.00G)	Atari ST	51109	3½"	399,–*	345,–	3990,–*
Update zu dBMAN (Vers. 3.00G)	Atari ST	51109U	3½"	98,–*	98,–	980,–*
WordStar/MailMerge	Atari ST	50106	3½"	199,–*	178,–	1990,–*
dBASE II	Atari ST	50306	3½"	348,–*	295,–	3490,–*
Update Protext 1.X auf 2.1	Atari ST	51644	3½"	30,–*	30,–	299,–*
Protext 2.1	Atari ST	51643	3½"	148,–*	132,–	1490,–*

* inkl. MwSt. Unverbindliche Preisempfehlung



Zeitschriften · Bücher
Software · Schulung

707215

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

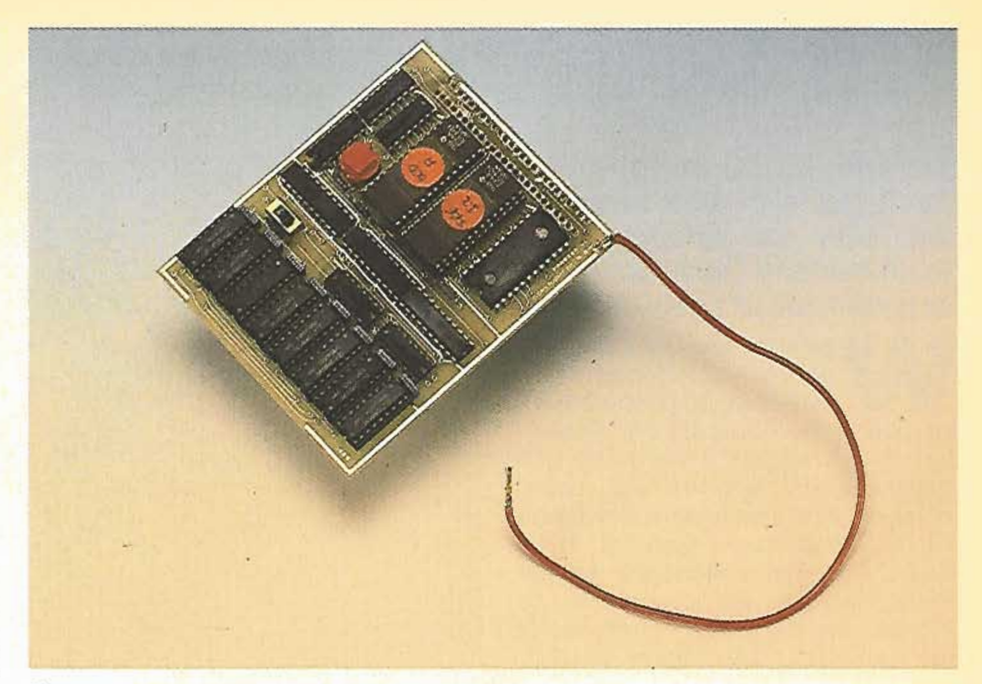
Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

freezter und gespeicherter Form wesentlich kürzer.

Der vierte Menüpunkt speichert ein eingefrorenes Programm nicht auf Diskette oder Kassette, sondern auf einer RAM-Disk. Auf der Freezerplatine sind zu diesem Zweck acht Sockel für RAM-Chips freigelassen, mit denen der Freezer in Schritten von 64 KByte auf 256 KByte aufgerüstet werden kann. Mit 64 KByte RAM auf der Platine hat man also den Speicher eines 130 XE zur Verfügung. Nach der Anleitung ist man damit jedoch nicht ganz 130-XE-kompatibel. Im 130 XE werden die zusätzlichen 64 KByte RAM in vier 16-KByte-Bänken in den Speicher eingeblendet. Der XE läßt sich nun so programmieren, daß der Mikroprozessor und der Bildschirmprozessor auf unterschiedliche Seiten zugreifen können. Falls ein Programm diese Betriebsart ausnutzt, wird es nicht ohne weiteres mit der RAM-Erweiterung des Turbo-Freezers XL zusammenarbeiten. Zum einen gibt es jedoch bisher kein kommerzielles Programm mit dieser Betriebsart. Zum anderen ist in der Anleitung sehr genau beschrieben, wie man ein Kabel im Inneren des Computers so an den Erweiterungsbus anlötet, daß das Umschaltsignal zur Verfügung steht. Doch zurück zum Freezer-Menü. »Save Ramdisk« speichert wie »Save extern« den Computerspeicher mit allen Registerwerten. Das Gegenstück dazu ist der Menüpunkt »Exec Ramdisk«, der aus der RAM-Disk ein Programm lädt, auftaut und startet.

Register eingefroren

Der sechste und letzte Menüpunkt läßt ahnen, welches Potential in dem Turbo-Freezer XL steckt. Wählt man den Menüpunkt »DOS/Debugger« an, hat man einen kompletten Monitor sowie ein komplettes DOS zur Verfügung. Mit dem Monitor lassen sich dann endlich die tiefsten Geheimnisse von Profiprogrammierern lüften. Für Leute, die beim Anblick von Programmen wie »Rescue on Fractalus« in andächtiges Staunen verfallen und bei denen sich im Kopf riesengroße, auf den Kopf gestellte Fragezeichen bilden, wenn sie gefragt werden, wie man so etwas programmiert, ist diese Funktion ideal. Ohne Probleme kann man nun ein in voller Aktion angehaltenes Programm untersuchen. Zusätzlich schaltet der Freezer automatisch das RAM unter dem Betriebssystem ein, damit man sich die dort stehenden Programme anschauen kann. Weiterhin ist der Freezer so konstruiert, daß das eigentliche Freezer-Programm, sozusagen das Betriebssystem des Freezers, nicht sichtbar ist und auch keinen Speicherplatz im Computer belegt. Ein Programm bemerkt also niemals den Freezer und hat somit auch keine Möglichkeit, Gegenmaßnahmen einzuleiten. Dem Freezer und dem Monitor ist jedes Programm schutzlos ausgeliefert. Nach Belieben läßt sich ein Programm disassemblieren, verändern oder entschützen. Für eigene Programme ist der Monitor das ultimative Entwicklungswerkzeug, da sich jedes Programm sofort beim Auftauchen eines Fehlers im Fehlerzustand untersuchen läßt.



Der Freezer Prototyp. Die Serienmodelle werden mit Lötstopmaske geliefert

Das eingebaute DOS stellt einen weiteren Knüller des Freezers dar. Ein Knopfdruck genügt, und es läßt sich mal eben schnell eine Diskette formatieren. Dabei gibt es jedoch einen kleinen negativen Kritikpunkt. Jedes normale DOS schreibt nach dem Formatieren einer Diskette auf die ersten drei Sektoren ein Programm, das auf dem Bildschirm »Boot Error« ausgibt, wenn von der Diskette gebootet wird und sich kein DOS darauf befindet. Der Freezer tut dies nicht, weshalb sich der Computer aufhängt, wenn man versucht, von einer Freezer-formatierten Diskette zu booten. Weitere Folgen hat dies keine, es ist halt nur nicht besonders schön.

Selbstverständlich werden Disketten in Single Density (90 KByte), Medium Density (130 KByte) und Double Density (180 KByte) unterstützt. Letztere allerdings nur mit einem aufgerüsteten Laufwerk. Neben den Befehlen zum Formatieren von Disketten stehen Kommandos zum Anzeigen des Inhaltsverzeichnisses einer Diskette, zum Schützen, Entschützen und Umbennen von Files sowie zum Laden und Speichern von Daten zur Verfügung.

Auf der Freezer-Platine ist neben dem EPROM mit der Betriebssoftware des Freezers noch ein weiterer Sockel frei. Gleichzeitig befindet sich auf der Platine über dem roten Knopf ein Schiebeschalter. Blättert man in der Anleitung, so wird der Sinn des freien Sockels schnell klar: Hier läßt sich ein EPROM mit dem Betriebssystem der alten 400er/800er Computer von Atari einsetzen. Es gibt einige Programme, die mit dem Betriebssystem der neuen Computer nicht funktionieren. Da kommt das alte Betriebssystem im ROM sehr recht, denn damit hat man quasi einen alten Atari Computer und ab sofort keine Probleme mehr mit inkompatibler Software.

Bei aller Euphorie hat der Freezer jedoch einen Nachteil: Er läßt sich nur am Atari 800XL betreiben, die Besitzer eines 130XE müssen vorerst noch auf eine Adapterplatine warten.

Das alte Betriebssystem sowie die RAM-Disk in beliebiger Ausbaustufe kann gleich mitgeordert werden. Dazu schreibt man an:

Gerald Engl
Kennwort: Turbo-Freezer XL
Bunsenstraße 13
8000 München 83

Der Freezer kostet 149 Mark, das alte Betriebssystem zusätzlich 10 Mark und jede 64-KByte-Stufe der RAM-Disk 20 Mark. Wenn man sich anschaut, was der Freezer alles leistet, und dann seinen Preis mit dem von Konkurrenzprodukten vergleicht, kann man nur sagen, daß der Freezer ein echter Hammer ist. (hf)

Ein Laufwerk wird fröhlich

Schneller, besser, komfortabler. Damit läßt sich kurz beschreiben, was das »Happy Drive Enhancement« aus der 1050-Diskettenstation herausholt.

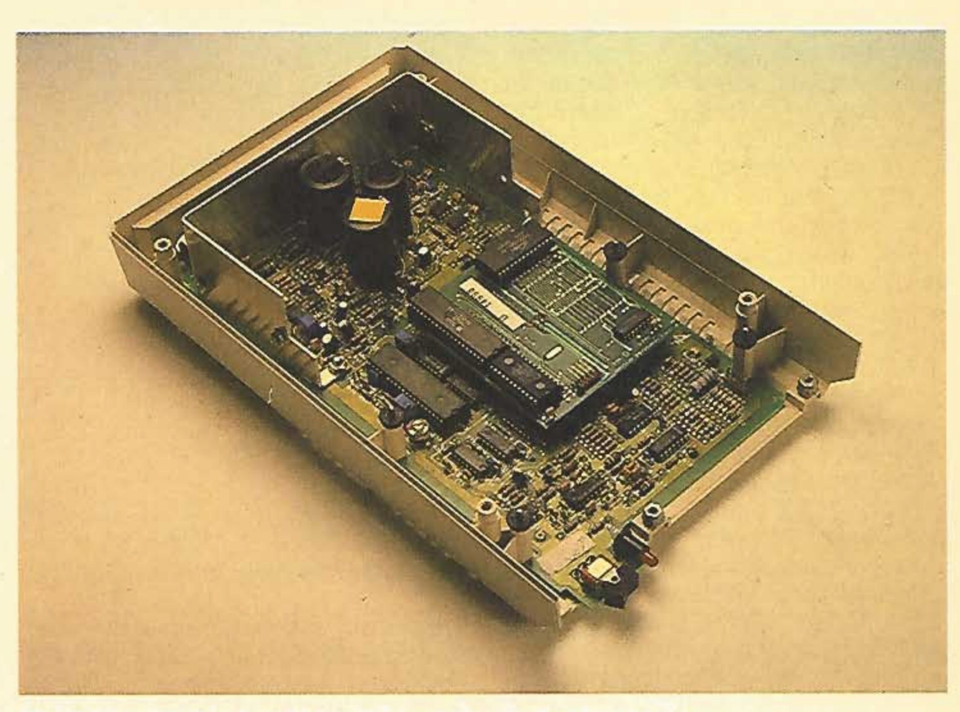
Was wäre ein Computer wie der Atari XL ohne eine Diskettenstation? Doch Disketten-Operationen bezahlt der Anwender oft mit langen Wartezeiten und umständlichem Kopierschutz. Diese Tatsache ließ die amerikanische Firma Happy Computers schon im Jahre 1982 auf den Gedanken kommen, ein Hardware-Utility auf den Markt zu bringen, mit der diese Nachteile für den Benutzer aufgehoben werden. Die Entwickler nannten ihre Erweiterung »Happy Drive Enhancement«. Diese Idee war von so durchschlagendem Erfolg, daß sie in Deutschland einige Nachahmer wie »Speedy 1050« oder »Turbo-Modul« nach sich zog. Doch an das Original reichten diese Nachfolger nicht heran. Ein Grund für uns, den »Ur-Speeder« etwas genauer anzusehen.

Kurze Ladezeiten

Das Happy Drive Enhancement, das den Platz des ROM und den des Prozessors in der Diskettenstation einnimmt, versetzt seinen Besitzer in die Lage, den Controller direkt zu programmieren und damit die Diskettenstation vielseitiger zu handhaben, als es normalerweise möglich ist.

Auf der aufzusteckenden Platine sind neben ein paar Logikbausteinen ein 8-KByte-EPROM, ein 8-KByte-RAM sowie eine 6502 CPU untergebracht – ein vollständiger kleiner Computer also. In dem RAM werden nicht nur die Übergabeparameter des Computers gespeichert, sondern es dient auch dazu, immer einen Track vollständig einzulesen. Dadurch wird eine geringere Belastung der Mechanik und vor allem eine erhebliche Geschwindigkeitssteigerung erreicht, da nachfolgende Sektoren schon zur Verfügung stehen und nicht mehr extra gelesen werden müssen.

Ein vom Anwender benutztes normales DOS merkt nichts von diesen Vorgängen, da sie alle von dem Happy Drive Enhancement gesteuert werden. Weiterhin ist es möglich, eigene Programme im Happy Drive Enhancement laufen zu lassen und so die Diskettenstation fest im Griff zu haben.



Eine kleine Platine im Laufwerk macht's möglich

Der Benutzer kann seine Disketten mit dem Enhancement für die Atari 1050 auf Double-Density formatieren, was ihm dann 176 KByte auf jeder Diskettenseite zur Verfügung stellt. Den Besitzern eines alten Laufwerks, der Atari 810, steht die doppelte Schreibdichte nicht zur Verfügung, da der dort eingebaute Controller dieses Format nicht erzeugen und verwalten kann. Wegen der unterschiedlichen Controller-Typen (bei der Atari 810 ist es der 1771, bei der Atari 1050 der 2793, beide von Western Digital) gibt es zwei verschiedene Happy Drive Enhancements, die Software ist jedoch identisch.

Ein weiterer Vorteil des Speeders: Der Benutzer ist nicht mehr auf die Übertragungsrate von 19200 Baud (übertragene Bits pro Sekunde) zwischen Computer und Diskettenstation angewiesen, sondern kann mit bestimmten Software-Einstellungen bis zu 70000 Baud erreichen. Damit wird aus dem monotonen Piep-Piep beim Laden und Speichern ein hohes Pfeifen.

Verzichtet man hierauf, kann man selbst in der normalen Betriebsart die Geschwindigkeit steigern, indem man ein DOS erzeugt, bei dem Verify-Write ausgeschaltet ist (mit POKE 1913,80). Das Happy Drive Enhancement führt diese Kontrolle schon selbständig durch.

Zusätzlich zu der Platine werden das »Happy Warp Speed Menü« und das »Warp Speed Dos« mitgeliefert, das mit dem Atari Dos 2.0 zusammenarbeitet.

Das Warp-Speed-Menü erscheint mittlerweile in der Version 7.1, die dem Benutzer neun Menüpunkte zur Verfügung stellt:

Drive Options:

Hier kann der Anwender

- sein Laufwerk wieder zu einem gewöhnlichen Atari-Diskettenlaufwerk ohne Beschleuniger machen, damit manche heiklen Original-Programme laufen. Sämtliche Funktionen des Happy Drive Enhancement sind dann abgeschaltet. Nur nach einem Aus- und wieder Anschalten wird das Atari-Laufwerk wieder »fröhlich«.
- das Happy Drive Enhancement initialisieren, um vorher eingestelle Optionen wieder aufzuheben.
- den »Drive-Select« außer Kraft setzen und Laufwerk 1 beispielsweise zu Laufwerk 2 machen.
- die Zeit bestimmen, in der das Laufwerk nach dem letzten Zugriff noch laufen soll, um bei einem eventuell erneuten Zugriff die Anlaufzeit des Motors zu sparen.
- auf schnelles Schreiben schalten.
- den U.S. Emulator installieren. Kopierprogramme beispielsweise, die diesen ansprechen, schreiben und lesen dann um ein Vielfaches schneller als ohne den Emulator.
- Laufwerk auf Write-Enable stellen.
- Laufwerk auf Write-Protect stellen.

Diagnostic:

Hier wird das »Happy Enhancement Diagnostic«-Programm nachgeladen.

Es folgt eine Auswahl von Test-Funktionen: RAM und ROM der Erweiterungen werden genauso überprüft wie Übertragungsrate, Anzahl der Umdrehungen pro Minute, Kopfpositionierung, Schreib- und Lesesicherheit und die zusätzliche Hardware-Option.

Sector Copier:

Ein Diskkopierer wird nachgeladen, mit dem Atari-Disketten auf sehr komfortable Art und Weise dupliziert werden können. So erlaubt es die Erweiterung hier, Start- und Endsektor des zu kopierenden Diskettenbereichs einzugeben.

Happy Backup:

Der Diskettenkopierer ist das Herzstück dieses Menüs. Dieses nachzuladende Programm kann allein oder mit Hilfe sogenannter »PDP-Files« fast alle kopiergeschützten Single-Density-Disketten duplizieren. PDP-kopierte Disketten laufen dann nur mit eingebauter Happy-Erweiterung.

Happy Compacter:

Nach dem Nachladen stößt der Anwender wieder in ein Menü, in dem es ihm möglich ist, mehrere Boot-Disketten auf einer zu sammeln. Tracks, die ein Programm nicht nutzt, werden für ein anderes freigehalten.

Multi Drive:

Bei diesem Kopierprogramm werden zwei Happy-Enhancement-Laufwerke benötigt. Dadurch entfällt das nervende dreimalige Auswechseln von Quell- und Zieldisketten.

Enable Tracer:

Der Tracer wird hier aktiviert. Jeden weiteren Track, der beim Diskettenzugriff angesprochen wird, merkt sich das Happy Drive Enhancement, und dieser kann nach erneutem Laden des »Happy Warp Speed Menü« abgefragt werden. Die Diskettenstation darf zwischendurch nicht ausgeschaltet werden, da sonst kein Tracen möglich wäre.

Display Trace:

Nur möglich, wenn vorher die Funktion »Enable Tracer« angewählt wurde. Es werden die Tracks 0 bis 39 angezeigt, die beim Booten benutzt wurden. Da beim Booten immer der Sektor 1 (Track 0) gelesen wird, und das Happy Warp Speed Menü nur Track 0 benutzt, ist das Aufzählen der gelesenen Tracks dem der vorherigen Boot-Diskette gleich. Deshalb ist das Happy Warp Speed Menü als Boot-Programm realisiert, als File könnte es auch noch andere Tracks belegen und so das Ergebnis verfälschen.

Insgesamt hinterläßt diese erste und vielseitigste Erweiterung einen hervorragenden Eindruck. Denn obwohl es andere und kostengünstigere Zusätze ähnlicher Art gibt, ist Happy Drive Enhancement das ausgereifteste Produkt. Außerdem unterstützen einige amerikanische und deutsche Softwarehersteller diesen Speeder mit ihren Programmen.

Mit diesen Programmen ist es möglich, sich ein völlig eigenes Format mit eigenem Kopierschutz zu erstellen. Mit der Atari 1050 kann man dann zum Beispiel ein Format in Medium-Density erzeugen, das jede 1050 ohne Probleme liest. Aber noch nicht einmal das Happy Drive könnte diese Diskette kopieren.

Das Happy-Drive-Enhancement ist für 149,95 Dollar plus 10 Dollar Versandkosten bei der Firma Happy-Computers, P.O. Box 1268, Morgan Hill, CA 95037 erhältlich.

Wer sich vor einer Bestellung in den USA scheut, kann sich an einen Atari-Händler wenden, muß dann aber mehr Geld ausgeben. Wie Sie eine Bestellung direkt in die USA richten, haben wir ausführlich in der Happy-Computer Ausgabe 7/87 beschrieben.

(Kjersten Waldheim/jg)

SPIELE-SPASS IST ANGESAGT!



Sonderheft 03/85: Spiele-Tests



Sonderheft 11: Spiele-Tests



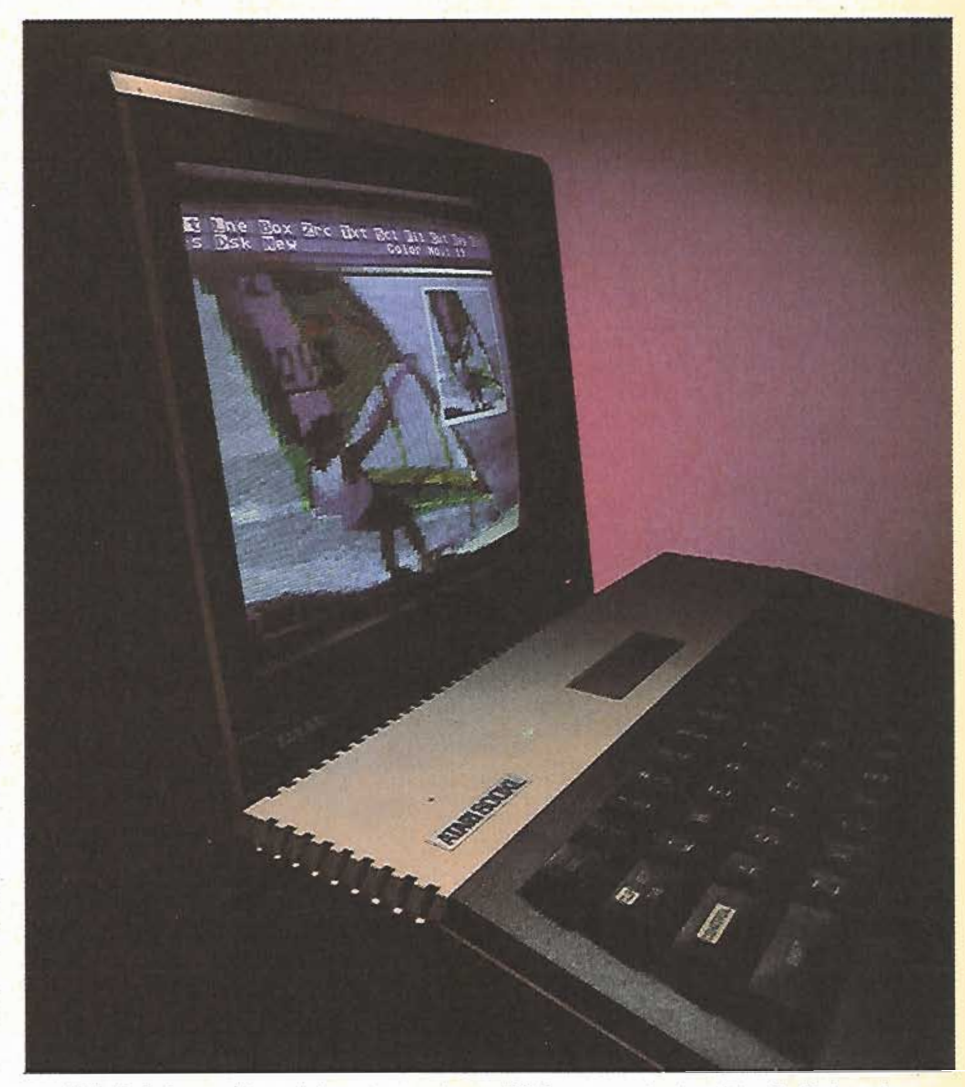
Sonderheft 17: Spiele-Tests

Achtung Spiele-Fans: Hier kommen insgesamt mehr als dreihundert Seiten voll mit Spiele-Tests, Marktübersichten, Grafik- und Musikprogrammen. Zusammen ergeben Sie ein Super-Nachschlagewerk: Geschicklichkeits-Spiele, Action-Spiele, Strategiespiele, Sport-Spiele …
vom C64 bis zum PC.

Nutzen Sie die Bestellmöglichkeit der Happy-Computer-Sonderhefte mit der eingehefteten Zahlkarte nach der Seite 98 in dieser Ausgabe.

Der XL treibt's ganz schön bunt

Mit ein bißchen Computerkenntnis, ein paar Assembler-Zeilen und einer guten Idee lassen sich aus den XL/XE-Computern viele Farben gleichzeitig herauskitzeln.



Eine der Stärken des Atari XL-Computers ist seine Farbenvielfalt. Er kann 16 Farben in 8 verschiedenen Helligkeiten darstellen, was zusammen 128 verschiedene Farben ergibt. Angesichts dieser Farbenvielfalt ist man begeistert, doch nach kurzer Zeit kommt die Ernüchterung: 128 Farbschattierungen stehen zwar zur Auswahl, in den einzelnen Grafikstufen sind aber nur maximal 4, in den Textstufen maximal 5 Farben gleichzeitig darstellbar. Es gibt zwar die Möglichkeit, über einen Display-List-Interrupt in jeder Bildschirmzeile die Farben umzuschalten. In vertikaler Richtung sind so beliebig viele Farben darstellbar, trotzdem hat man in einer horizontalen Zeile immer noch nur 4 beziehungsweise 5 Farben zur Verfügung.

Beim Atari gibt es außerdem 3 weitere Grafikstufen, in denen ein Bildpunkt eine Zeile hoch und vier Punkte breit ist. In der Grafikstufe 9 sind 16 Graustufen einer Farbe darstellbar und in Grafikstufe 11 sind es 16 Farben einer Helligkeit. Bei Grafikstufe 10 sind 7 Farben frei definierbar.

Um 256 Farben gleichzeitig auf dem Bildschirm darstellen zu können, ist folgende Überlegung notwendig: Das Fernseh- oder Monitorbild wird 50mal in der Sekunde neu gezeichnet. Wenn jetzt zwischen einem Bild mit Grauwerten und einem Bild mit Farbwerten im Takt des Bildschirmaufbaus hin- und hergeschaltet wird, kann das Auge des Betrachters die Bilder nicht mehr auseinanderhalten und mischt die Grau- und die Farbwerte zu einer Farbe zusammen. Auf diese Weise lassen sich aus 16 Grauwerten und 16 Farbwerten 256 Farben zurechtmischen.

Das Verfahren hat jedoch einige Nachteile, die hier nicht verschwiegen werden sollen. Zum ersten tritt durch das rasche Hin- und Herschalten zwischen den beiden Bildern ein unangenehmer Nebeneffekt auf: Das Bild flackert sehr stark. Dieser Effekt trat auch bei einigen alten Adventurespielen von Scott Adams auf. Nach den ersten drei Spielen sah er jedoch ein, daß diese Art der Bilderzeugung viel zu anstrengend für den Betrachter ist. Ab dem vierten Spiel verwendete er dann wieder die normale Art der Bilderzeugung. Ohne Bedeutung ist das Flackern jedoch, wenn man Bilder zeichnet und diese dann vom Bildschirm abfotografiert (siehe Bilder in diesem Beitrag und in der Inhaltsangabe dieser Ausgabe). Zum zweiten lassen sich bei diesem Verfahren, bedingt durch die Auflösung der beiden verwendeten Grafikstufen, horizontal nur 80 Punkte darstellen. Und zum dritten muß man einen nicht unerheblichen programmtechnischen Aufwand betreiben, um das Bild darzustellen. Im Computer muß nämlich ein Programm fest installiert sein, das die Bilder umschaltet. Zusätzlich nehmen diese Programmroutinen sowie die beiden Bilder Speicherplatz weg.

Dem Flackern läßt sich mit einem Kniff beikommen: Wie bei einem Fernsehbild verschiebt man jedes zweite übertragene Bild gegenüber dem ersten um eine Zeile. In jeder geraden Zeile werden dann die Grauwerte dargestellt und in jeder ungeraden Zeile die Farbwerte.

Damit man ein 256-Farben-Bild ausgeben kann, braucht man ein Programm, daß die Bilder hin- und herschaltet, sowie zwischen Grau- und Farbwerten wechselt. Zusätzlich braucht man zwei Display Lists, die das Farbbild gegenüber dem Helligkeitsbild um eine Zeile verschieben. Das Listing »COL256.BIN« stellt dieses Programm dar. Das Programm steht im Speicher ab Adresse 1536

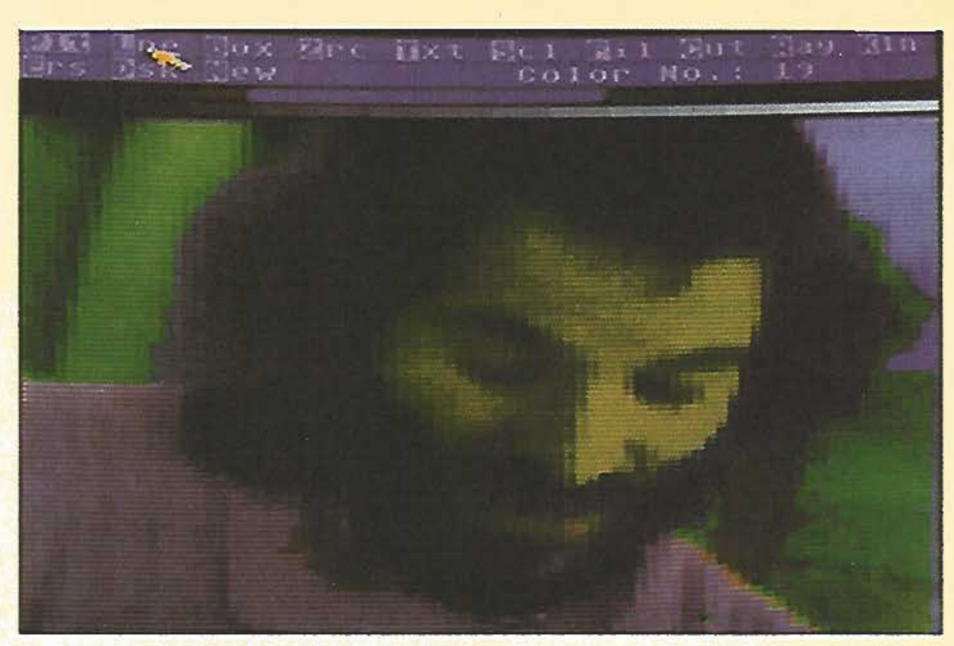
($600) und belegt 104 Byte. Es ist so geschrieben, daß es unter Turbo-Basic mit dem Befehl »BLOAD "D:COL256.BIN"« geladen wird. Das Programm »FARB256« enthält alle Programmschritte, die nötig sind, um einen 256-Farben-Bildschirm zu erzeugen. Am besten speichern Sie das Programm mit »LIST "D:FARB 256.TXT"« auf Ihre Diskette, damit Sie es mit dem Befehl »ENTER« in eigene Programme einbinden können. Das Programm springt man unter Turbo-Basic einfach mit »EXEC FARB256_ INIT« an, worauf die Display-Lists installiert und das COL256.BIN-Programm geladen werden.

Dann schaltet man die Grafik mit folgendem Befehl an:

```
D=USR($0600,GTIAMode)
```

Für die Variable »D« läßt sich jede andere im Programm nicht benützte Variable einsetzen. Sie wird von der Funktion zwar nicht verwendet, muß aber trotzdem angegeben werden, da der »USR«-Befehl eine Variable erwartet. Solange keine Player-Missile-Grafik verwendet wird, setzt man für GTIAMode normalerweise eine 64 ein, was Grafikmode 9 entspricht. Ansonsten schauen Sie bitte im Beitrag über die Hardwareregister der Atari 8-Bit-Computer unter dem Register »PRIOR« nach, welche Werte Sie einsetzen müssen. Nachdem die Grafik nun aktiviert ist, muß der Bildschirmspeicher gegen unbeabsichtigtes Überschreiben durch ein Basic-Programm geschützt werden. Dazu verwendet man folgende Befehlsfolge:

```
POKE 106,120
```



Mit »Paint256« erzeugtes, digitalisiertes Bild

Will man jetzt Grafiken zeichnen, so muß die gleiche Grafik in beiden Bildern gezeichnet werden, einmal mit dem Helligkeitsbild und das zweite Mal mit dem Farbbild. Dazu legt man erst einmal die Farbe und die Helligkeit fest. So ergibt:

```
COL = 3
LUM = 6
```




ein dunkles Rot. Dann teilt man dem Computer über die Speicherstelle 88 mit, wo sein Bildschirmspeicher steht. Alle Grafikbefehle beziehen sich jetzt auf diesen Bildschirmspeicher. Als erstes zeichnet man das Helligkeitsbild.

```
DPOKE 88,AH
POKE 87,9
```

Anschließend setzt man den Helligkeitswert und führt den Grafikbefehl aus.

```
COLOR LUM
PLOT 0;0:DRAWTO 79,191
```

Dann setzt man den Punkt. Die gleiche Prozedur wiederholt man mit dem Farbwert.

```
DPOKE 88,AF
COLOR COL
PLOT 0,0:DRAWTO 79,191
```

Auf diese Weise lassen sich in jedes Programm Grafiken mit 256 Farben einbauen. Da unter Basic mit installierter Grafik nur noch rund 16 KByte Speicher zur Verfügung stehen, sollte man Programme unbedingt compilieren. Dadurch gewinnt man zusätzliche 4 KByte Speicher. Bevor man in irgendeiner Weise den seriellen Port benutzt, sollte man die Grafik wieder abschalten. Dies geschieht mit dem Befehl:

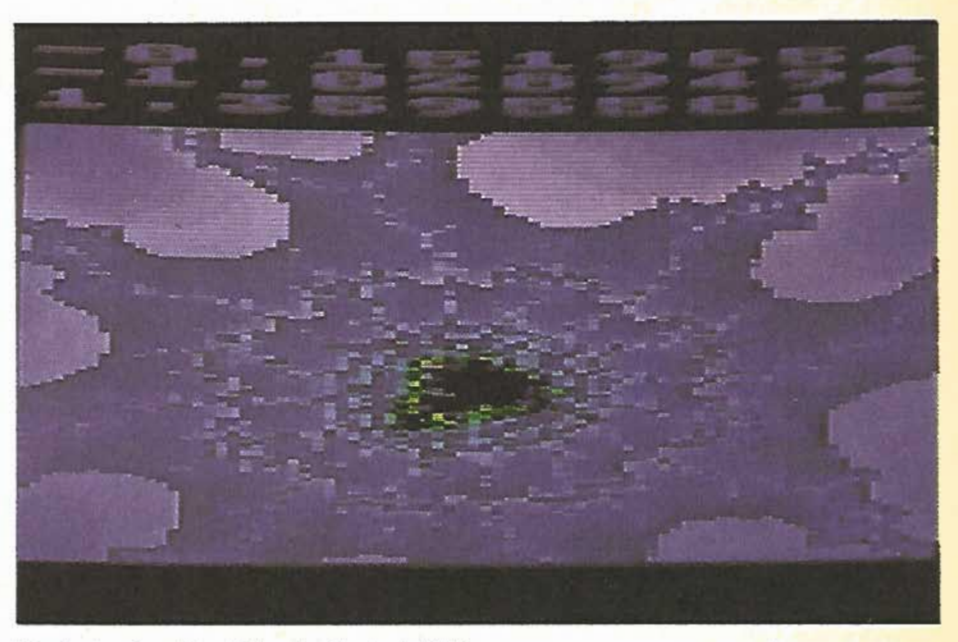
```
D=USR($0600,0)
```

Ansonsten hat der Computer nicht mehr genügend Rechenzeit zur Verfügung und hängt sich auf. Als Beispiel haben wir auf Seite 35 ein Programm abgedruckt, das alle 256 Farben gleichzeitig auf den Bildschirm bringt. Davon läßt sich auch sehr schön der Aufbau eines 256-Farben-Programms zeigen.

Nach der Arbeit kommt das Vergnügen

Zusätzlich zu den Routinen, die die 256 Farben ermöglichen, haben wir zwei Anwenderprogramme abgedruckt, die diese Routinen benutzen. Zum einen haben wir ein Malprogramm beigefügt. Es ist in Turbo-Basic geschrieben und benötigt die beiden Programme COL256.BIN sowie »DLST.256«. Diese Programme müssen sich zusammen mit »PAINT256« auf der gleichen Diskette befinden.

PAINT256 stellt verschiedene Funktionen zur Verfügung. Diese präsentieren sich nach dem Starten des Programms in der obersten Bildschirmzeile. Darunter folgen zwei schmale Zeilen mit 16 Farb- und 16 Helligkeitswerten, und unter diesen befindet sich die Zeichenfläche, auf der horizontal 80 und vertikal 192 Punkte darstellbar sind. Mit dem Pfeil, den man mit einem Joystick in Port 1 des Computers bewegt, werden dann die einzelnen Funktionen ausgewählt. Um eine Farbe festzulegen, fährt man mit dem Pfeil auf den entsprechenden Farb- oder Helligkeitsbalken und drückt den Feuerknopf. Zusätzlich läßt sich die Farbe mit den Pfeiltasten auf der Tastatur bestimmen. Die Aufwärts-/Abwärts-Pfeile verändern die Farbe und die Links-/Rechts-Pfeile bestimmen die Helligkeit einer Farbe. Die Nummer der ausgewählten Farbe wird in der zweiten Menüzeile über der Zeichenfläche bei »Color No.:« angezeigt.



Kunterbuntes Mandelbrot mit »Apfel256«

Mit dem Pfeil wählt man auch die einzelnen Zeichenfunktionen aus. Eine gewählte Funktion wird dann vollständig invers dargestellt. Folgende Funktionen stehen zur Verfügung, wobei sich fast alle Funktionen mit <ESC> abbrechen lassen, wenn sie einmal versehentlich angewählt wurden.

Plt (Plot): Wenn der Knopf gedrückt ist, setzt das Programm an der entsprechenden Stelle auf dem Bildschirm einen Punkt.

Lne (Line) zieht eine Linie von dem Punkt, an dem das erstemal der Knopf betätigt wurde, zu dem Punkt des zweiten Knopfdrucks.

Box (Box): Der erste Knopfdruck legt den Eckpunkt eines Kastens fest. Wird dann der Pfeil bewegt, so wird von dem Eckpunkt zu der neuen Pfeilposition ein Kasten gezogen. Beim zweiten Knopfdruck wird der Kasten gezeichnet.

Zrc (Circle): Beim ersten Knopfdruck wird der Mittelpunkt eines Kreises bestimmt. Beim zweiten wird ein Kreis mit dem Radius vom Abstand des Mittelpunkts zur Position des Pfeils gezeichnet.

Txt (Text): Nach dem der Knopf betätigt wurde, läßt sich ein Text mit maximal 10 Zeichen Länge eingeben. Mit <BACKSPACE> wird eine fehlerhafte Eingabe korrigiert. <ESC> löscht den gesamten Text. Mit <RETURN> wird der Text übernommen und mit dem Joystick positioniert. Durch Knopfdruck wird der Text in das Bild eingefügt.

Scl (Scale): Mit dieser Funktion wird der Ausschnitt eines Bildes vergrößert oder verkleinert. Dabei dürfen sich die Bereiche auch überlappen. Da der Computer sehr viel Rechenzeit verbraucht, wenn er einen Bildschirmausschnitt transformiert, wird zwischendurch der Bildschirm ausgeschaltet. Durch diesen Trick hat der Computer mehr Zeit zur Verfügung.

Fil (Fill): Eine schwarze Fläche wird mit der gewählten Farbe ausgefüllt.

Cut (Cut): Diese Funktion schneidet einen rechteckigen Ausschnitt aus und fügt ihn an anderer Stelle wieder ein.

Ray (Rays): Ausgehend von einem gemeinsamen Punkt werden Linien gezogen. Indem man im Menü eine andere Funktion auswählt, beendet man diese Funktion.

Cln (continue Lines): Es werden fortlaufende Linien gezogen. Der Endpunkt einer Linie ist gleichzeitig der Startpunkt der nächsten Linie. Auch diese Funktion wird erst durch eine andere Funktion beendet.

Ers (Erase): Der Pfeil verwandelt sich in ein schraffiertes Zeichen, das eine Art Radiergummi darstellt. Es löscht Flächen aus einem Bild.



Superbase – das relationale Datenbank-System für den Atari ST

Superbase vereint als erstes Programm einer neuen Generation von Datenbank-Systemen sowohl eine neuartige, äußerst benutzerfreundliche Bedienung mit Pull-down-Menüs, Fenstern und Maussteuerung als auch die enorme Leistungsfähigkeit einer relationalen Dateiverwaltung.

Einfacher Datenbank-Aufbau

Mit den leichtverständlichen Pull-down-Menüs und Kontrollfeldern legen Sie in Minuten eine komplette Datenbank an. Sie können eine bereits festgelegte Struktur jederzeit ändern, ohne Ihre Daten zu zerstören.

Verwaltung der Daten

Superbase zeigt Ihre Daten auf verschiedene Arten an, beispielsweise als Tabelle oder als Formular. Sind Index und Felder selektiert, können Sie Ihre Daten wie bei einem Videorecorder anzeigen lassen. Schneller Vorlauf, Rücklauf, Pause und Stop – ein Recorder ist nicht einfacher zu bedienen. Ein einzigartiges Filtersystem wählt beliebige Datenkategorien aus, mit denen Sie dann arbeiten können.

Die Stärken von Superbase

Das Festlegen von Übersichten und zusammenhängende Abfragen über mehrere verknüpfte Dateien sind auch bei verschiedenen Sortierkriterien kein Problem. Daten anderer Datenbanken oder Anwenderprogramme im ASCII-Format lassen sich ebenfalls problemlos verarbeiten. Binden Sie Daten in Ihre Textverarbeitung ein, oder bilden Sie aus verschiedenen Dateien eine neue Datenbank!

Datenbank mit Bildern

Superbase bietet neben den gängigen Datenbank-Funktionen die Möglichkeit, Bilder und Grafiken darzustellen und zu verwalten. Einzigartigen Grafik-Datenbanken oder Dia-Shows steht also nichts im Wege.

Wer braucht Superbase?

Die Anwendungsmöglichkeiten sind nahezu unbegrenzt.

Hier einige Beispiele:

Geschäftliches	Professionelle *Anwendungen*
Lagerbestand Fakturierung Registratur Versandlisten Verwaltung Adressen	Design Fotografie Journalismus Sammlungen Forschung Ausbildung

Übrigens: Superbase gibt es auch für Amiga, Schneider PC und IBM-PCs und Kompatible

Leistungsumfang

Die Software: ● bis zu 16 Millionen Datensätze pro Datei ● maximal 999 Indizes pro Datei ● Speicherkapazität pro Datei, Anzahl der geöffneten Dateien, Anzahl der Dateien und Anzahl der Felder pro Datensatz: abhängig von der Disketten- bzw. Festplattenkapazität.

Die Daten: ● Text, Daten, numerische Felder und externe Dateien ● Überprüfung bei der Eingabe ● Formelfelder ● Kalender der Jahre 1–9999, verschiedene Datumsformen ● verschiedene Zahlenformate bei 13stelliger Genauigkeit ● Datenschutz per Paßwort

Die Ausgaben: ● das Programm beherrscht einen flexiblen Etikettendruck und produziert übersichtliche Listen mit dem Reportgenerator ● bis zu 255 Spalten ● mit Titel, Datum und Seitenzahl ● Datensatz-Zähler, Durchschnitt, Zwischen- und Endergebnis ● Ausgabe von mehreren Dateien auf Bildschirm, Drucker, Diskette oder neuer Datei ● mehrspaltiger Etikettendruck mit variablem Format ● Speicherung der Ausgabe- und Abfrage-Formate zur späteren Verwendung ● vielfältige Sortierkriterien

Hardware-Anforderung

● Atari ST 512 Kbyte ● 3½"-Diskettenlaufwerk ● 80-Zeichen-Monitor

Lieferumfang: ● 3½"-Diskette ● Handbuch deutsch

Best.-Nr. 51641

DM 249,–* (sFr 199,–/öS 2490,–*)

* inkl. MwSt. Unverbindliche Preisempfehlung



Zeitschriften · Bücher
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

705218

Mandelbrot-Ausschnitte			
Realteil	Imaginärteil	Seitenlänge	Ausschnitt
-2	-1.25	2.5	Gesamtansicht
-0.225	-1.125	0.25	Zweig
-0.74591	0.11196	1.43E-3	Spirale
-0.1765625	-1.0484375	0.03125	kleines Mandelbrot

Dsk (Disk): Es erscheint ein neues Menü, über das verschiedene Diskettenfunktionen ausgeführt werden:
- **Save** (Save): Das Bild wird auf der Diskette gespeichert, wobei ein Bild 123 Sektoren belegt.
- **Load** (Load): Sämtliche Files mit dem Zusatz ».PIC« werden in der obersten Zeile angezeigt. Ein Knopfdruck lädt das Bild. Indem man den Joystick bewegt, wird das nächste File angezeigt.
- **Frmt** (Format): Formatiert die Diskette in Laufwerk 1. Dazu muß die Taste <Y> gedrückt werden. Alle anderen Tasten brechen die Funktion sofort ab.
- **Dir** (Directory): Zeigt sämtliche Files mit dem Zusatz ».PIC« an. Durch Joystickbewegung wird das nächste File aufgeführt, Knopfdruck beendet die Funktion.
- **EDir** (extended Directory): Dies funktioniert wie die Directory-Funktion. Im Gegensatz dazu werden jedoch alle Files ausgegeben.
- **Bye** (Bye): Das Programm kehrt wieder ins Hauptmenü zurück.
- **Quit** (Quit): Das Programm wird beendet, und man befindet sich wieder im Turbo-Basic.

New (New): Nach Drücken der Taste <Y> wird das Bild gelöscht.

Der Regenbogen über dem Mandelbrot

Das zweite Programm berechnet die gute alte Mandelbrotmenge, bei dieser Version allerdings in 256 Farben. Das Prinzip der Mandelbrotberechnung soll hier nicht weiter erläutert werden. Gerade in der letzten Zeit haben sich wissenschaftliche Magazine und Computerzeitschriften mit Beiträgen über die Mandelbrotmenge geradezu überschlagen, so daß die Berechnung der Bilder bekannt sein sollte. Wer trotzdem wissen möchte, wie Mandelbrot funktioniert, und sich allgemein für Fractals interessiert, sollte sich das Buch »The Beauty of Fractals« von H.-O. Pleitgen und P. H. Richter anschauen. Es ist zwar in englischer Sprache verfaßt, bietet aber soviel Stoff zu diesem Thema, daß es fast als »Bibel« der Mandelbrotmenge und deren Nebenarten bezeichnet werden kann. Nebenbei enthält es viele phantastische Bilder. Das Buch ist im Springer-Verlag Heidelberg erschienen und kostet 78 Mark.

Zurück zum Mandelbrotprogramm. Gerade bei Bildern aus der Mandelbrotmenge sind viele Farben wichtig, da die Bilder sonst einen Großteil ihres Reizes verlieren. Auch dieses Programm ist in Turbo-Basic geschrieben und sollte mit dem Prüfsummer eingegeben werden. Nach dem Starten meldet sich das Programm mit folgendem Menü:

Laden Bild: Das Programm verlangt eine dreistellige Nummer des zu ladenden Bildes. Anschließend wird das Bild mit dem Dateinamen »APFELXXX.PIC« geladen.

Speicher Bild: Wie bei »Laden Bild«, nur daß bei dieser das Bild vom Speicher auf Diskette übertragen wird.

Berechne Bild: Nach Eingabe der Werte für die untere linke Ecke, der Seitenlänge und der Berechnungstiefe beginnt das Programm einen Ausschnitt aus der Mandelbrotmenge zu berechnen. Dieser Ausschnitt ist wie bei den Originalprogrammen auf Großcomputern quadratisch. Die Berechnungstiefe sollte größer oder gleich 256 sein, da sonst nicht alle darstellbaren Farben ausgenutzt werden.

Ausschnitt vom Bild: Diese Funktion läßt sich nur aufrufen, wenn sich ein Bild im Speicher des Computers befindet. Das Programm schaltet auf das Bild um, und es erscheinen zwei Rahmen auf dem Bildschirm, die die rechte obere sowie die linke untere Ecke eines neu zu berechnenden Bildausschnitts repräsentieren. Der Ausschnitt wird mit dem Joystick über das Bild bewegt. Ein Druck auf den Joystickknopf verändert die Ausschnittsgröße. Bei Druck auf die <START>-Taste fragt das Programm nach der Rechentiefe. Anschließend wird der neue Mandelbrotausschnitt berechnet.

Während die Berechnung des Bildes erfolgt, läßt sich der Bildschirm mit der <START>-Taste abschalten. Dadurch arbeitet der Computer doppelt bis dreimal so schnell. Mit der <SELECT>-Taste wird der Bildschirm wieder angeschaltet.

Directory: Das Inhaltsverzeichnis des Diskettenlaufwerks wird angezeigt.

<SELECT>-Taste: Das berechnete Bild erscheint. Falls kein Bild im Speicher ist, bleibt der Bildschirm schwarz.

<START>-Taste: Es wird wieder zum Menü zurückgeschaltet.

Im unteren Bereich des Bildschirms werden die Bildparameter des berechneten beziehungsweise geladenen Bildes angezeigt. Ein Bild nimmt auf der Diskette zwischen 123 und 124 Sektoren in Anspruch, je nachdem, wie viele Stellen die Bildwerte aufweisen. Damit Sie, ohne viel in der Mandelbrotmenge herumzusuchen, Bilder mit dem Programm »APFEL256« berechnen können, haben wir für Sie Werte interessanter Ausschnitte zusammengestellt.

Vielleicht wissen Sie weitere interessante Anwendungen für den 256-Farben-Zusatz. Schreiben Sie uns doch Ihre Vorschläge für weitere Anwendungen. Wir freuen uns über jede Anregung.

(Stephan Gerle/hf)

Steckbrief	
Programm:	Paint256
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Diskette

Steckbrief	
Programm:	Apfel256
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Diskette

```
Programmname :COL256.BIN
Länge        :112 Bytes
-----------------------------------

0000:FF FF 00 06 67 06 68 68 <9A>
0008:68 C9 00 F0 20 85 D0 A9 <C9>
0010:FF 8D 66 06 A2 06 A0 36 <B2>
0018:A9 07 20 5C E4 A9 57 8D <7D>
0020:00 02 A9 06 8D 01 02 A9 <4B>
0028:C0 8D 0E D4 60 A2 E4 A0 <5F>
0030:62 A9 07 20 5C E4 A9 60 <8A>
0038:8D 0E D4 60 A5 D0 29 7F <10>
0040:85 D0 A2 00 A0 7C AD 66 <9D>
0048:06 49 80 8D 66 06 30 04 <FD>
0050:A2 00 A0 78 8E 30 02 8C <79>
0058:31 02 4C 62 E4 48 A5 D0 <6A>
0060:49 80 85 D0 8D 0A D4 8D <1B>
0068:1B D0 68 40 00 00 00 00 <2F>
```

Listing. »COL256.BIN«

```
Programmname :DLST.256
Länge        :1192 Bytes
-----------------------------------
0000:70 00 00 42 10 9E 00 02 <59>
0008:90 0F 0F 0F 0F 90 0F 0F <CA>
0010:0F 0F 10 CF 10 80 CF 38 <9E>
0018:A0 CF 60 80 CF 88 A0 CF <AC>
0020:B0 80 CF D8 A0 CF 00 81 <7C>
0028:CF 28 A1 CF 50 81 CF 78 <38>
0030:A1 CF A0 81 CF C8 A1 CF <20>
0038:F0 81 CF 18 A2 CF 40 82 <F8>
0040:CF 68 A2 CF 90 82 CF B8 <A2>
0048:A2 CF E0 82 CF 08 A3 CF <93>
0050:30 83 CF 58 A3 CF 80 83 <44>
0058:CF A8 A3 CF D0 83 CF F8 <2D>
0060:A3 CF 20 84 CF 48 A4 CF <4F>
0068:70 84 CF 98 A4 CF C0 84 <D0>
0070:CF E8 A4 CF 10 85 CF 38 <9B>
0078:A5 CF 60 85 CF 88 A5 CF <A3>
0080:B0 85 CF D8 A5 CF 00 86 <7F>
0088:CF 28 A6 CF 50 86 CF 78 <CA>
0090:A6 CF A0 86 CF C8 A6 CF <93>
0098:F0 86 CF 18 A7 CF 40 87 <03>
00A0:CF 68 A7 CF 90 87 CF B8 <F4>
00A8:A7 CF E0 87 CF 08 A8 CF <49>
00B0:30 88 CF 58 A8 CF 80 88 <53>
00B8:CF A8 A8 CF D0 88 CF F8 <3E>
00C0:A8 CF 20 89 CF 48 A9 CF <8A>
00C8:70 89 CF 98 A9 CF C0 89 <DF>
00D0:CF E8 A9 CF 10 8A CF 38 <B4>
00D8:AA CF 60 8A CF 88 AA CF <C2>
00E0:B0 8A CF D8 AA CF 00 8B <8E>
00E8:CF 28 AB CF 50 8B CF 78 <5E>
00F0:AB CF A0 8B CF C8 AB CF <92>
00F8:F0 8B CF 18 AC CF 40 8C <96>
0100:CF 68 AC CF 90 8C CF B8 <46>
0108:AC CF E0 8C CF 08 AD CF <9F>
0110:30 8D CF 58 AD CF 80 8D <D9>
0118:CF A8 AD CF D0 8D CF F8 <88>
0120:AD CF 20 8E CF 48 AE CF <5B>
0128:70 8E CF 98 AE CF C0 8E <F7>
0130:CF E8 AE CF 10 8F CF 38 <C6>
0138:AF CF 60 8F CF 88 AF CF <0F>
0140:B0 8F CF D8 AF CF 00 90 <96>
0148:CF 28 B0 CF 50 90 CF 78 <81>
0150:B0 CF A0 90 CF C8 B0 CF <A0>
0158:F0 90 CF 18 B1 CF 40 91 <AC>
0160:CF 68 B1 CF 90 91 CF B8 <1B>
0168:B1 CF E0 91 CF 08 B2 CF <96>
0170:30 92 CF 58 B2 CF 80 92 <69>
0178:CF A8 B2 CF D0 92 CF F8 <5D>
0180:B2 CF 20 93 CF 48 B3 CF <96>
0188:70 93 CF 98 B3 CF C0 93 <F2>
0190:CF E8 B3 CF 10 94 CF 38 <58>
0198:B4 CF 60 94 CF 88 B4 CF <CE>
01A0:B0 94 CF D8 B4 CF 00 95 <A5>
01A8:CF 28 B5 CF 50 95 CF 78 <92>
01B0:B5 CF A0 95 CF C8 B5 CF <00>
01B8:F0 95 CF 18 B6 CF 40 96 <32>
01C0:CF 68 B6 CF 90 96 CF B8 <F4>
01C8:B6 CF E0 96 CF 08 B7 CF <91>
01D0:30 97 CF 58 B7 CF 80 97 <78>
01D8:CF A8 B7 CF D0 97 CF F8 <A6>
01E0:B7 CF 20 98 CF 48 B8 CF <55>
01E8:70 98 CF 98 B8 CF C0 98 <FC>
01F0:CF E8 B8 CF 10 99 CF 38 <F4>
01F8:B9 CF 60 99 CF 88 B9 CF <7D>
0200:B0 99 CF D8 B9 CF 00 9A <9D>
0208:CF 28 BA CF 50 9A CF 78 <24>
0210:BA CF A0 9A CF C8 BA CF <AC>
0218:F0 9A CF 18 BB CF 40 9B <42>
0220:CF 68 BB CF 90 9B CF B8 <4E>
0228:BB CF E0 9B CF 08 BC CF <00>
0230:30 9C CF 58 BC CF 80 9C <7A>
0238:CF A8 BC CF D0 9C CF F8 <78>
0240:BC CF 20 9D CF 48 BD CF <83>
0248:70 9D CF 98 BD CF C0 9D <10>
0250:CF E8 BD 41 70 00 00 42 <16>
0258:10 9E 00 02 90 0F 0F 0F <D1>
0260:0F 90 0F 0F 0F 0F 90 CF <45>
0268:10 A0 CF 38 80 CF 60 A0 <59>
0270:CF 88 80 CF B0 A0 CF D8 <50>
0278:80 CF 00 A1 CF 28 81 CF <E1>
0280:50 A1 CF 78 81 CF A0 A1 <45>
0288:CF C8 81 CF F0 A1 CF 18 <3B>
0290:82 CF 40 A2 CF 68 82 CF <A5>
0298:90 A2 CF B8 82 CF E0 A2 <53>
02A0:CF 08 83 CF 30 A3 CF 58 <71>
02A8:83 CF 80 A3 CF A8 83 CF <55>
02B0:D0 A3 CF F8 83 CF 20 A4 <F2>
02B8:CF 48 84 CF 70 A4 CF 98 <DB>
02C0:84 CF C0 A4 CF E8 84 CF <AA>
02C8:10 A5 CF 38 85 CF 60 A5 <68>
02D0:CF 88 85 CF B0 A5 CF D8 <25>
02D8:85 CF 00 A6 CF 28 86 CF <2C>
02E0:50 A6 CF 78 86 CF A0 A6 <E4>
02E8:CF C8 86 CF F0 A6 CF 18 <D1>
02F0:87 CF 40 A7 CF 68 87 CF <64>
02F8:90 A7 CF B8 87 CF E0 A7 <DE>
0300:CF 08 88 CF 30 A8 CF 58 <F3>
0308:88 CF 80 A8 CF A8 88 CF <8B>
0310:D0 A8 CF F8 88 CF 20 A9 <2A>
0318:CF 48 89 CF 70 A9 CF 98 <ED>
0320:89 CF C0 A9 CF E8 89 CF <1F>
0328:10 AA CF 38 8A CF 60 AA <0E>
0330:CF 88 8A CF B0 AA CF D8 <89>
0338:8A CF 00 AB CF 28 8B CF <9E>
0340:50 AB CF 78 8B CF A0 AB <FE>
0348:CF C8 8B CF F0 AB CF 18 <E7>
0350:8C CF 40 AC CF 68 8C CF <92>
0358:90 AC CF B8 8C CF E0 AC <69>
0360:CF 08 8D CF 30 AD CF 58 <88>
0368:8D CF 80 AD CF A8 8D CF <42>
0370:D0 AD CF F8 8D CF 20 AE <78>
0378:CF 48 8E CF 70 AE CF 98 <FE>
0380:8E CF C0 AE CF E8 8E CF <76>
0388:10 AF CF 38 8F CF 60 AF <E1>
0390:CF 88 8F CF B0 AF CF D8 <01>
0398:8F CF 00 B0 CF 28 90 CF <18>
03A0:50 B0 CF 78 90 CF A0 B0 <05>
03A8:CF C8 90 CF F0 B0 CF 18 <FA>
03B0:91 CF 40 B1 CF 68 91 CF <50>
03B8:90 B1 CF B8 91 CF E0 B1 <82>
03C0:CF 08 92 CF 30 B2 CF 58 <99>
03C8:92 CF 80 B2 CF A8 92 CF <81>
03D0:D0 B2 CF F8 92 CF 20 B3 <3F>
03D8:CF 48 93 CF 70 B3 CF 98 <13>
03E0:93 CF C0 B3 CF E8 93 CF <31>
03E8:10 B4 CF 38 94 CF 60 B4 <A5>
03F0:CF 88 94 CF B0 B4 CF D8 <93>
03F8:94 CF 00 B5 CF 28 95 CF <1F>
0400:50 B5 CF 78 95 CF A0 B5 <94>
0408:CF C8 95 CF F0 B5 CF 18 <95>
0410:96 CF 40 B6 CF 68 96 CF <8E>
0418:90 B6 CF B8 96 CF E0 B6 <90>
0420:CF 08 97 CF 30 B7 CF 58 <A3>
0428:97 CF 80 B7 CF A8 97 CF <3E>
0430:D0 B7 CF F8 97 CF 20 B8 <0F>
0438:CF 48 98 CF 70 B8 CF 98 <35>
0440:98 CF C0 B8 CF E8 98 CF <6F>
0448:10 B9 CF 38 99 CF 60 B9 <5F>
0450:CF 88 99 CF B0 B9 CF D8 <A1>
0458:99 CF 00 BA CF 28 9A CF <25>
0460:50 BA CF 78 9A CF A0 BA <1B>
0468:CF C8 9A CF F0 BA CF 18 <26>
0470:9B CF 40 BB CF 68 9B CF <75>
0478:90 BB CF B8 9B CF E0 BB <A7>
0480:CF 08 9C CF 30 BC CF 58 <3C>
0488:9C CF 80 BC CF A8 9C CF <6D>
0490:D0 BC CF F8 9C CF 20 BD <26>
0498:CF 48 9D CF 70 BD CF 98 <46>
04A0:9D CF C0 BD CF E8 9D 41 <8F>
04A8:00 00 00 00 00 00 00 00 <87>
```

Listing. »DLST.256«

```
10 --                                         <YW>
20 REM Paint 256  (c) by Stephan Gerle &
 Francois Paquet-Durand                      <PL>
30 REM Tel.: 02506/7493 (Stephan) ; 0252
6/2120 (Francois)                            <BF>
40 --                                         <YZ>
1020 EXEC DECLARE:EXEC MENUE_INIT:EXEC A
N                                            <PS>
1040 POKE 709,12:POKE 710,98                 <RW>
1070 # START                                 <NG>
1071 POP :TRAP #FEHLER                       <SB>
1072 POKE 87,%0:DPOKE 88,ME:PLOT %3,%3:P
OSITION 32,%1:? COL*16+LUM;"▲▲";             <EV>
2000 --                                       <VF>
2010 DO                                      <TQ>
2020 EXEC WAIT_KEY                           <JW>
2030 CHALT=CH                                <XG>
2040 IF T>%0                                 <FL>
2050 T=T&127:D=%0                            <IL>
2060 IF T=42 THEN LUM=LUM+(LUM<15):D=%1      <KU>
2070 IF T=43 THEN LUM=LUM-(LUM>%0):D=%1      <UQ>
2080 IF T=61 THEN COL=COL+(COL<15):D=%1      <TK>
2090 IF T=45 THEN COL=COL-(COL>%0):D=%1      <DU>
2100 IF D=%1 THEN GO# START                  <OR>
2110 T=T&95                                  <OB>
2120 IF T=80 THEN CH=%1                      <RP>
2130 IF T=76 THEN CH=%2                      <UY>
2140 IF T=66 THEN CH=%3                      <VN>
2150 IF T=90 THEN CH=4                       <NS>
2160 IF T=84 THEN CH=5                       <QC>
2170 IF T=83 THEN CH=6                       <QP>
2180 IF T=70 THEN CH=7                       <PT>
2190 IF T=67 THEN CH=8                       <TN>
2200 IF T=82 THEN CH=9                       <RY>
2210 IF T=75 THEN CH=10                      <CS>
2211 IF T=69 THEN CH=11                      <FE>
2212 IF T=68 THEN CH=12                      <FT>
2213 IF T=78 THEN CH=13                      <HF>
2220 ELSE                                    <TF>
2230 IF Y<-23                                <KG>
2240 CH=X DIV 8+%1                           <XH>
2250 ELSE                                    <TO>
2251 IF Y<-14 AND X<22                       <JE>
2252 CH=10+X DIV 8+%1                        <MK>
2253 ELSE                                    <UA>
2260 IF Y<-8 AND Y>-12                       <GB>
2270 COL=X DIV 5:GO# START                   <AN>
2280 ELSE                                    <TX>
2290 IF Y<-2 AND Y>-7                        <SE>
2300 LUM=X DIV 5:GO# START                   <DU>
2310 ENDIF                                   <SQ>
2320 ENDIF                                   <ST>
2340 ENDIF                                   <SZ>
2341 ENDIF                                   <TD>
2350 ENDIF                                   <TC>
2360 IF CHALT<>CH                            <TA>
2370 POS=CHALT*4-%3+ME                       <LU>
2380 POKE POS,PEEK(POS)&127                  <MK>
2390 POKE POS+%1,PEEK(POS+%1)&127            <IO>
2400 POS=CH*4-%3+ME                          <FV>
2410 POKE POS,PEEK(POS)!128                  <IC>
2420 POKE POS+%1,PEEK(POS+%1)!128            <DI>
2430 ENDIF                                   <SY>
2440 IF Y>-%1 AND STRIG(%0)=%0 OR CH>11      <LR>
2450 EXEC SAFE:XA=X:YA=Y:EXEC HELL:D=%0      <HI>
2460 ON CH EXEC PLOT,LNE,BOX,CRC,TXT,SCL
,FIL,CUT,RAY,RAY,RUB,DSK,NEW                 <JM>
2470 ENDIF                                   <TK>
```

Listing. »Paint256«

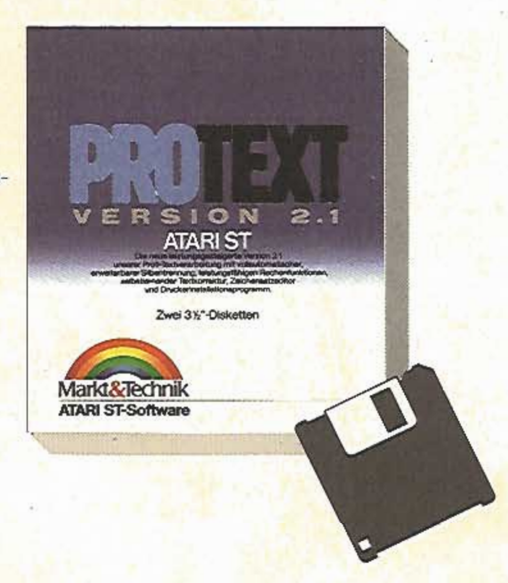
ATARI

PROTEXT für die ATARI ST

Version 2.1

Die professionelle deutsche Textverarbeitung mit vollautomatischer Silbentrennung und integrierter Tabellenkalkulation.

Ab sofort erhalten Sie PROTEXT in einer leistungsstärkeren Version – zum alten Preis. Als Besitzer der Version 1.x (1.0–1.3) haben Sie die Möglichkeit, kostengünstig ein Update zu erhalten.
PROTEXT ist ein leicht bedienbares, Maus-unterstütztes Textprogramm mit hoher Leistungsfähigkeit. Eingebaute Hilfefunktionen ermöglichen auch dem Anfänger, die gesamte Leistungsfähigkeit dieser professionellen Software zu nutzen. Das Programm erlaubt die direkte Eingabe und Änderung aller Attribute wie **Fettschrift**, Unterstreichen, *Kursivschrift*, Breitschrift, Hoch- und Tiefstellen. Der Text ist ohne besondere Ausgabe auf dem Bildschirm sofort formatiert sichtbar, so wie er auch auf dem Drucker ausgegeben wird. Der Zeichensatz ist frei definierbar. Alle Positionen im Text (vom verfügbaren Speicherplatz abhängig) können mit entsprechenden Funktionen sehr schnell aufgesucht werden.
Die schon bekannten Leistungsmerkmale: vollautomatische Silbentrennung, der Text wird auf dem Bildschirm genauso ausgegeben, wie er später ausgedruckt wird (WYSIWYG-Prinzip), komfortable Funktionsübersicht, universelle Druckeranpassung, Verketten von Texten, Archiv-Funktionen, komfortable, programmierbare Rechenfunktionen, selbstlernende Textkorrektur mit Grundwortschatz, Textlupe, Zeichensatzanzeige.



Neue Features:
Block markieren jetzt direkt mit Maus möglich, Maskenmodus, Zugriff auf im Desktop installierte Accessories, Text direkt drucken, Ganzseitenübersicht, erweiterbare, verbesserte Silbentrennung, optimierte Druckerinstallation, geschwindigkeitsgesteigerte Formatierroutine und vieles mehr.

Hardware-Anforderung:
- Atari 260 ST, 520 ST, 520 ST+, 1040 ST,
- Monochrommonitor
- beliebiger Drucker

Bestell-Nr. 51643

Für nur DM 148,–*

PROTEXT Update 1.x auf 2.1:

Bestell-Nr. 51644

Für nur DM 30,–*

(gegen Einsendung der Originaldiskette oder Registrierungskarte)

* inkl. MwSt. Unverbindliche Preisempfehlung

	Version	Best.-Nr.	Format	Preis DM	sFr	öS
WordStar/MailMerge	Atari ST	50106	3½"	199,-*	178,-	1990,-*
dBASE II	Atari ST	50306	3½"	348,-*	295,-	3490,-*
Update zu Protext 1.x	Atari ST	51644	3½"	30,-*	30,-	299,-*
Protext 2.1	Atari ST	51643	3½"	148,-*	132,-	1490,-*

* inkl. MwSt. Unverbindliche Preisempfehlung



Zeitschriften · Bücher
Software · Schulung

706217

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

```
2480 POKE 764,255                          <GY>
2490 LOOP                                  <ZE>
2990 --                                    <WY>
3000 PROC DECLARE                          <DF>
3010 DIM LEER$(256),MAUS$(11),NAME$(16),
UNDO$(7680)                                <EK>
3031 PMB=120:POKE 54279,PMB:PMB=PMB*256+
1839                                       <EC>
3040 CH=%1:CHALT=CH:XX=%1:COL=%1:LUM=%3:
UN=ADR(UNDO$)                              <KW>
3090 POKE 707,255:POKE 53277,%2:MAUS$="{
CTL ,}{CTL ,}@p{SHIFT =}x<.{CTL G}{CTL B
}{CTL ,}":MOVE ADR(LEER$),PMB,256          <CF>
3120 AH=$8010:AF=$A010:ME=AH+$1E00         <XR>
3140 OPEN #%1,4,%0,"D:DLST.256":BGET #%1
,$7800,596:BGET #%1,$7C00,596:CLOSE #%1    <PU>
3170 BLOAD "D:COL256.BIN"                  <OW>
3180 POKE AF,%0:MOVE AF,AF+%1,$1E00:MOVE
 AF,AH,$1E00                               <GC>
3210 POKE 106,120:GRAPHICS %0:POKE 82,%0   <YV>
3290 ENDPROC                               <PJ>
3300 PROC PARTSAFE                         <OV>
3310 IF YA<Y:YY=YA*40:ELSE :YY=Y*40:ENDI
F :MOVE UN+YY,AH+YY,ABS(YA-Y)*40+40        <CA>
3390 ENDPROC                               <PL>
3530 PROC MENUE                            <CV>
3540 DPOKE 88,ME:PLOT %3,%3:POSITION %0,
%0:MOVE ADR(LEER$),ME,60:POKE 87,%0        <YR>
3570 ENDPROC                               <PJ>
3580 PROC HELL                             <HQ>
3590 DPOKE 88,AH:POKE 87,9:COLOR LUM       <EP>
3600 ENDPROC                               <DQ>
3610 PROC FARBE                            <PQ>
3620 DPOKE 88,AF:POKE 87,9:COLOR COL       <HK>
3630 ENDPROC                               <OZ>
3640 PROC AN                               <HN>
3650 D=USR($0600,65):POKE 559,58           <RE>
3660 ENDPROC                               <PI>
3670 PROC AUS                              <AP>
3680 POKE 559,%0:D=USR($0600,%0):PAUSE %
3                                          <RZ>
3690 ENDPROC                               <PR>
3700 PROC WAIT_KEY                         <NW>
3710 REPEAT                                <ES>
3720 EXEC MAUS                             <LQ>
3730 UNTIL PEEK(764)<255 OR STRIG(%0)=%0   <OC>
3740 IF PEEK(764)<255:GET T:ELSE :T=%0:E
NDIF                                       <QN>
3750 ENDPROC                               <PH>
3760 PROC SAFE                             <DP>
3770 MOVE AH,UN,$1E00                      <RY>
3780 ENDPROC                               <PQ>
3790 PROC UNSAFE                           <TZ>
3800 MOVE UN,AH,$1E00                      <OF>
3810 ENDPROC                               <OX>
3820 PROC MENUE_INIT                       <VC>
3830 EXEC MENUE:POKE 752,1                 <MJ>
3840 ? "Plt▲Lne▲Box▲Zrc▲Txt▲Scl▲Fil▲Cut▲
Ray▲Kln▲Ers▲Dsk▲New▲"                      <GJ>
3851 POSITION 21,%1:? "Color▲No.:▲"        <EW>
3870 CH=%1:POKE 87,9:DPOKE 88,ME+80        <YC>
3880 FOR I=%0 TO 15                        <MW>
3890 COLOR I:TEXT I*5,%0,"▲"               <XS>
3910 NEXT I                                <FM>
3920 ENDPROC                               <PC>
4120 PROC PLOT                             <TC>
4130 WHILE STRIG(%0)=%0 AND Y>-%1          <QN>
4140 EXEC HELL:PLOT X,Y:EXEC FARBE:PLOT
X,Y:EXEC MAUS                              <PC>
4190 WEND                                  <UK>
4200 ENDPROC                               <OJ>
4210 PROC FIL                              <SS>
4220 IF COL>%0 AND LUM>%0                  <IT>
4230 EXEC HELL:PAINT X,Y:EXEC FARBE:PAIN
T X,Y                                      <DH>
4240 ENDIF                                 <SZ>
4260 ENDPROC                               <PB>
4270 PROC LNE                              <UT>
4280 WHILE STRIG(%0) AND Y>-%1 AND PEEK(
764)<>28                                   <VN>
4290 PLOT XA,YA:DRAWTO X,Y:EXEC PARTSAFE
:EXEC MAUS                                 <MA>
4310 WEND                                  <TQ>
4320 IF Y>-%1 AND PEEK(764)<>28 THEN PLO
T XA,YA:DRAWTO X,Y:EXEC FARBE:PLOT XA,YA
:DRAWTO X,Y:PAUSE %3                       <SQ>
4340 ENDPROC                               <OX>
4350 PROC BOX                              <AI>
4400 WHILE STRIG(%0) AND Y>-%1 AND PEEK(
764)<>28                                   <UT>
4420 PLOT XA,YA:DRAWTO XA,Y:DRAWTO X,Y:D
RAWTO X,YA:DRAWTO XA,YA                    <YP>
4430 EXEC PARTSAFE:EXEC MAUS               <RS>
4440 WEND                                  <UB>
4450 IF Y>-%1 AND PEEK(764)<>28            <LL>
4470 PLOT XA,YA:DRAWTO XA,Y:DRAWTO X,Y:D
RAWTO X,YA:DRAWTO XA,YA                    <ZE>
4472 IF D=%0                               <AO>
4480 EXEC FARBE:DRAWTO XA,Y:DRAWTO X,Y:D
RAWTO X,YA:DRAWTO XA,YA                    <SI>
4482 ENDIF                                 <TX>
4490 ENDIF                                 <TS>
4500 ENDPROC                               <OP>
4510 PROC CRC                              <RI>
4520 WHILE STRIG(%0) AND Y>%0 AND PEEK(7
64)<>28                                    <GG>
4530 CIRCLE XA,YA,ABS(XA-X),ABS(YA-Y):EX
EC UNSAFE:EXEC MAUS                        <MI>
4550 WEND                                  <UG>
4560 IF Y>-%1 AND PEEK(764)<>28            <LQ>
4580 CIRCLE XA,YA,ABS(XA-X),ABS(YA-Y):EX
EC FARBE:CIRCLE XA,YA,ABS(XA-X),ABS(YA-Y
)                                          <FF>
4590 ENDIF                                 <TU>
4600 ENDPROC                               <OR>
4610 PROC TXT                              <JW>
4620 L=10:G=%1:POS=%0:NAME$="":IF Y<184
THEN EXEC INPUT                            <LZ>
4630 WHILE STRIG(%0) AND Y>%0 AND Y<184
AND PEEK(764)<>28                          <GS>
4640 TEXT X,Y,NAME$:MOVE UN+Y*40,AH+Y*40
,320:EXEC MAUS                             <GX>
4660 WEND                                  <UL>
4670 IF Y<%0 OR PEEK(764)=28:EXEC UNSAFE   <QA>
4680 ELSE :TEXT X,Y,NAME$:EXEC FARBE:TEX
T X,Y,NAME$:WHILE STRIG(%0)=%0:WEND :END
IF                                         <LO>
4690 ENDPROC                               <PS>
4700 PROC SCL                              <VS>
4710 D=%1:EXEC BOX:P1=XA:P2=YA:P3=X:P4=Y   <EA>
4712 IF PEEK(764)<>28 AND X>XA AND Y>YA    <AX>
4720 EXEC UNSAFE:REPEAT :EXEC MAUS:UNTIL
 STRIG(%0)=%0:XA=X:YA=Y:EXEC HELL          <EJ>
4730 PAUSE 10:EXEC BOX:P5=XA:P6=YA:P7=X:
P8=Y                                       <WW>
4732 IF PEEK(764)<>28 AND X>XA AND Y>YA    <BD>
4740 EXEC UNSAFE:EXEC SAFE:EXEC AUS        <GK>
4750 Z1=(P1-P3)/(P5-P7):Z2=(P2-P4)/(P6-P
8)                                         <IU>
4762 FOR I=%0 TO %1                        <BR>
4770 FOR I1=P5 TO P7                       <WC>
4780 FOR I2=P6 TO P8                       <XY>
4810 DPOKE 88,UN:LOCATE P1+(I1-P5)*Z1,P2
+(I2-P6)*Z2,Z:COLOR Z:DPOKE 88,AH+I*$200
0                                          <RK>
4820 DPOKE 88,AH+I*$2000:PLOT I1,I2        <JS>
4850 NEXT I2                               <CZ>
4860 NEXT I1                               <CQ>
4870 MOVE AF,UN,7680                       <KT>
4880 NEXT I                                <GG>
4882 EXEC AN                               <EM>
4884 ELSE :EXEC UNSAFE:ENDIF               <IZ>
4886 ELSE :EXEC UNSAFE:ENDIF               <JH>
4890 ENDPROC                               <PW>
4900 PROC DSK                              <WK>
4920 EXEC MENUE:? "Save▲Load▲Frmt▲Dir▲▲E
Dir▲Bye▲▲Quit"                             <GJ>
4930 REPEAT                                <FD>
4940 EXEC WAIT_KEY:D=%0:T=T&95             <AL>
4950 IF T=%0                               <FY>
```

Listing. »Paint256« (Fortsetzung)



```
4951 IF Y<-23 THEN D=X DIV 10+%1                <YQ>
4960 ELSE                                       <UH>
4970 IF T=83 THEN D=%1                          <LY>
4980 IF T=76 THEN D=%2                          <NW>
4990 IF T=70 THEN D=%3                          <MB>
5000 IF T=68 THEN D=4                           <IP>
5010 IF T=69 THEN D=5                           <JZ>
5020 IF T=66 THEN D=6                           <JN>
5030 IF T=81 THEN D=7                           <IZ>
5040 ENDIF                                      <SW>
5060 EXEC MENUE:IF D<6 THEN ON D EXEC SA
VE,LAD,FORMAT,NDIR,EDIR:D=6                     <EK>
5070 EXEC MENUE_INIT                            <RD>
5080 UNTIL D>5                                  <RE>
5090 IF D=7 THEN EXEC AUS:GRAPHICS %0:PO
KE 53251,%0:END                                 <AC>
5100 ENDPROC                                    <OI>
5110 PROC RUB                                   <YH>
5120 MAUS$="{CTL ,}{ESC CTL >}U+U+U+{ESC
 CTL >}{CTL ,}{CTL ,}":POKE 87,15               <CL>
5130 WHILE Y>%0 AND PEEK(764)=255               <IU>
5140 WHILE STRIG(%0)=%0 AND Y>%1                <LS>
5150 EXEC MAUS:COLOR %0                         <UZ>
5160 DPOKE 88,AH:TEXT X*%2,Y-%1,"▲"             <BP>
5180 DPOKE 88,AF:TEXT X*%2,Y-%1,"▲"             <AP>
5200 WEND                                       <TM>
5210 EXEC MAUS                                  <LF>
5220 WEND                                       <TS>
5230 MAUS$="{CTL ,}{CTL ,}@p{SHIFT =}x<.
{CTL G}{CTL B}{CTL ,}"                          <JK>
5240 ENDPROC                                    <OW>
5250 PROC NEW                                   <AG>
5270 EXEC MENUE:? "▲▲Clear▲screen▲(Yes/N
o)▲?":EXEC WAIT_KEY                             <SV>
5290 IF T=89                                    <SC>
5300 POKE AF,%0:MOVE AF,AF+%1,$1E00:MOVE
 AF,AH,$1E00                                    <FK>
5330 ENDIF                                      <SZ>
5340 EXEC MENUE_INIT                            <RA>
5350 ENDPROC                                    <PB>
5360 PROC FORMAT                                <ZV>
5370 EXEC MENUE:? "O.k.to▲format▲disk▲in
▲drive▲1?▲(Yes/No)"                             <WP>
5380 EXEC WAIT_KEY                              <KO>
5390 IF T=89                                    <SE>
5400 EXEC AUS:XIO 254,#%1,%0,%0,"D1":EXE
C AN                                            <TZ>
5430 ENDIF                                      <TB>
5440 ENDPROC                                    <PA>
5450 PROC NDIR                                  <LC>
5460 LEER$="D:*.PIC":L=10:EXEC DIR:LEER$
="{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CT
L ,}{CTL ,}"                                    <MU>
5490 ENDPROC                                    <PP>
5500 PROC EDIR                                  <GU>
5510 LEER$="D:*.*":L=13:EXEC DIR:LEER$="
{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}"            <KJ>
5540 ENDPROC                                    <PC>
5550 PROC DIR                                   <VP>
5560 REPEAT                                     <FF>
5570 EXEC AUS:CLOSE #%1:OPEN #%1,6,%0,LE
ER$                                             <QM>
5590 REPEAT                                     <FO>
5610 EXEC AUS:INPUT #%1,NAME$:SL=11             <LI>
5612 IF NAME$(5)<>"FREE▲SECTORS"                <RL>
5620 POSITION 10,%0                             <DF>
5640 FOR I=%3 TO 10                             <KE>
5650 IF NAME$(I,I)="▲" THEN SL=SL-%1            <DR>
5660 NEXT I                                     <FX>
5670 IF NAME$(%2,%2)="▲":? NAME$(%3,L)          <EQ>
5680 ELSE :? NAME$(%1,15):ENDIF                 <JQ>
5690 EXEC AN                                    <EE>
5700 WHILE STICK(%0)=15 AND PEEK(764)=25
5 AND STRIG(%0):WEND                            <WW>
5710 IF PEEK(764)<255 THEN GET T                <FP>
5712 ENDIF                                      <TJ>
5720 UNTIL T=155 OR STRIG(%0)=%0 OR NAME
$(%2,%2)<>"▲" OR T=27                           <GW>
5730 UNTIL STRIG(%0)=%0 OR T=27 OR T=155        <UD>
5740 CLOSE #%1                                  <HX>
5750 ENDPROC                                    <PJ>
5760 PROC MAUS                                  <QE>
5770 A=STICK(%0)                                <KC>
5780 X=X+XX*(A&8=%0)-(A&4=%0)                   <QO>
5790 Y=Y+(A&%2=%0)-(A&%1=%0)                    <RD>
5800 IF Y>191 THEN Y=191                        <IN>
5810 IF Y<-32 THEN Y=-32                        <WD>
5820 IF X>79 THEN X=79                          <ZJ>
5830 IF X<%0 THEN X=%0                          <QU>
5850 POKE 53251,X*%2+48:MOVE ADR(MAUS$),
PMB+Y,11                                        <TC>
5860 ENDPROC                                    <PO>
5870 # FEHLER                                   <NB>
5890 CLOSE #%1:EXEC MENUE:EXEC AN               <XP>
5900 ? "▲▲▲Fehler▲Nr.:▲";ERR;".▲▲Taste/K
nopf▲"                                          <UC>
5910 EXEC WAIT_KEY:EXEC MENUE_INIT              <PQ>
5930 GO# START                                  <JB>
5940 PROC INPUT                                 <PL>
5950 IF NOT (G) THEN EXEC MENUE:POSITION
 2,0:? "Save▲as?":NAME$(%1)=""                  <FC>
5960 REPEAT                                     <FN>
5970 IF G:MOVE UN+Y*40,AH+Y*40,320:TEXT
X,Y,NAME$                                       <DL>
5980 ELSE :POSITION 11,0:? NAME$;"▲":END
IF                                              <ZW>
5990 GET T:D=T&127                              <VG>
6010 IF (D>-%1 AND D<27) OR (D>31 AND D<
125)                                            <ZN>
6020 POS=POS+(POS<L):NAME$(POS,POS)=CHR$
(T)                                             <UY>
6040 ENDIF                                      <SX>
6050 IF T=126 AND POS=%0 THEN NAME$=""          <UI>
6060 IF T=126 AND POS>%0 THEN POS=POS-%1
:NAME$(POS+%1)="▲"                              <TZ>
6070 UNTIL T=155 OR T=27                        <RQ>
6072 IF T=27 THEN POKE 764,28                   <KE>
6080 ENDPROC                                    <PF>
6090 PROC LAD                                   <OK>
6100 EXEC NDIR                                  <GI>
6110 IF T<>27 OR NAME$(%2,%2)<>"▲"              <GJ>
6120 NAME$(%1,%2)="D:":NAME$(SL)=".PIC"         <QH>
6130 EXEC AUS:OPEN #%1,4,%0,NAME$               <CO>
6140 BGET #%1,AH,$1E00:BGET #%1,AF,$1E00
:CLOSE #%1:EXEC AN                              <OC>
6170 ENDIF                                      <TI>
6190 ENDPROC                                    <PK>
6200 PROC SAVE                                  <KQ>
6210 G=%0:L=8:POS=%0:EXEC INPUT                 <ZM>
6220 IF T<>27                                   <RT>
6230 -MOVE ADR(NAME$(%1)),ADR(NAME$(%3))
,8                                              <OU>
6240 NAME$(%1,%2)="D:":NAME$(POS+%3)=".P
IC"                                             <TO>
6250 EXEC AUS:OPEN #%1,8,%0,NAME$               <GO>
6260 BPUT #%1,AH,$1E00:BPUT #%1,AF,$1E00
:CLOSE #%1:EXEC AN                              <UK>
6290 ENDIF                                      <TQ>
6300 ENDPROC                                    <ON>
7000 PROC CUT                                   <AV>
7010 D=%1:EXEC BOX:EXEC UNSAFE:P2=Y:XA=X
A DIV %2:P1=ABS(X DIV %2-XA):EXEC SAFE:X
X=%2                                            <YJ>
7020 IF Y>YA AND X>XA                           <GE>
7100 REPEAT                                     <EH>
7110 EXEC UNSAFE                                <OT>
7120 FOR I=YA TO P2                             <LO>
7122 IF (Y+I)*40-P2*40<%0 THEN POKE 764,
28:EXIT                                         <EM>
7130 MOVE XA+UN+I*40,X DIV %2+AH+(Y+I)*4
0-P2*40-P1+%1,P1                                <UV>
7140 NEXT I                                     <FJ>
7150 EXEC MAUS                                  <LR>
7160 UNTIL STRIG(%0)=%0 OR PEEK(764)=28         <JB>
7170 IF PEEK(764)<>28                           <FM>
7180 MOVE AF,UN,7680                            <KL>
7190 FOR I=YA TO P2                             <MJ>
7200 MOVE XA+UN+I*40,X DIV %2+AF+(Y+I)*4
0-P2*40-P1+%1,P1                                <SC>
7210 NEXT I                                     <FC>
```

Listing. »Paint256« (Fortsetzung)

ATARI

Spitzen-Software für ATARI ST

WordStar 3.0 mit MailMerge

Dieses Standardprogramm mit weltweitem Erfolg bietet Ihnen bildschirmorientierte Formatierung, deutschen Zeichensatz und integrierte Hilfstexte.
Lieferumfang: Zwei 3½-Zoll-Disketten mit CP/M-Emulator und WordStar/MailMerge-Dateien, Originalhandbuch von MicroPro, Beschreibung der ATARI-spezifischen Version.
Hardware-Anforderung: ATARI-ST-Computer, 80-Zeichen-Monitor, mindestens ein 3½-Zoll-Diskettenlaufwerk, Drucker mit Centronics-Schnittstelle.
Best.-Nr. 50106
DM 199,–* (sFr 178,–/öS 1990,–*)



MicroPro WordStar 3.0
mit MailMerge für den ATARI ST
3½"-Format

dBASE II 2.41

Das bedienerfreundliche Datenbanksystem dBASE II beinhaltet eine eigene Programmiersprache für die Erstellung individueller Programme. dBASE II wurde unter dem Betriebssystem GEM-TOS den ATARI-ST-Computern angepaßt und läßt sich komfortabel über Pull-down-Menüs mit der Maus steuern.
Lieferumfang: 3½-Zoll-Diskette, Originalhandbuch von Ashton-Tate, Beschreibung der ATARI-spezifischen GEM-Version.
Hardware-Anforderung: ATARI-ST-Computer mit Betriebssystem im ROM (260 ST, 520 ST, 520 ST+, 1040 ST), ein 3½-Zoll-Diskettenlaufwerk, Drucker mit Centronics-Schnittstelle.
Best.-Nr. 50306
DM 348,–* (sFr 295,–/öS 3490,–*)





unter GEM-TOS
3½"-Diskette für ATARI 260 ST, 520 ST, 520 ST+ und 1040 ST

Diese Markt&Technik-Software erhalten Sie in den Fachabteilungen der Warenhäuser, bei Ihrem Computerfachhändler, im Buchhandel oder direkt beim Verlag gegen Vorauskasse. Fragen Sie auch nach dem neuen Gesamtverzeichnis Frühjahr/Sommer '87. Oder fordern Sie es direkt beim Verlag an.

Dazu die weiterführende Literatur:





WordStar für ATARI ST

Das Buch für den mühelosen Einstieg in die Praxis der Textverarbeitung mit WordStar.
Best.-Nr. 90208, ISBN 3-89090-208-1

dBASE II für die ATARI ST

Zu einem Weltbestseller unter den Datenbanksystemen das klassische Einführungs- und Nachschlagewerk.
Best.-Nr. 90206, ISBN 3-89090-206-5

Jedes Buch kostet DM 49,–
(sFr 45,10/öS 382,20)
Erhältlich bei Ihrem Buch- oder Computerfachhändler.

* inkl. MwSt.
Unverbindliche Preisempfehlung

	Version	Best.-Nr.	Format	Preis DM	sFr	öS
WordStar/MailMerge	Atari ST	50106	3½"	199,-*	178,-	1990,-*
dBASE II	Atari ST	50306	3½"	348,-*	295,-	3490,-*
Update Protext 1.x auf 2.1	Atari ST	51644	3½"	30,-*	30,-	299,-*
Protext 2.1	Atari ST	51643	3½"	148,-*	132,-	1490,-*

* inkl. MwSt. Unverbindliche Preisempfehlung



Zeitschriften · Bücher
Software · Schulung

707211

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

```
7220 ELSE                                      <TK>
7230 EXEC UNSAFE                               <PB>
7240 ENDIF                                     <TC>
7250 ENDIF                                     <TF>
7260 XX=%1                                     <QE>
7290 ENDPROC                                   <PN>
7300 PROC RAY                                  <AR>
7310 REPEAT                                    <EO>
7320 EXEC LNE:EXEC HELL:EXEC SAFE:IF CH=
10 THEN XA=X:YA=Y                              <WB>
7350 UNTIL PEEK(764)=28 OR Y<%1                <NH>
7390 ENDPROC                                   <PP>
```

Listing. »Paint256« (Schluß)

```
1000 REM Mandelbrotmenge/Apfelmaennchen        <BD>
1010 REM in 256 Farben.                        <KC>
1020 --                                        <VK>
1030 REM Voraussetzung:                        <RJ>
1040 REM Atari XL/XE                           <EA>
1050 REM Floppy (Turbo Basic XL)               <LF>
1060 REM Grafikerweiterung fuer 256 Farb
en                                             <EY>
1070 REM (COL256.BIN auf Diskette in Lau
fwerk 1)                                       <WX>
1080 --                                        <WC>
1090 REM (c) by Stephan Gerle                  <GN>
1100 --                                        <VG>
1110 EXEC FARB256_INIT                         <NH>
1120 GRAPHICS %0                               <YG>
1130 DIM LU$(12),RO$(12),FILE$(20),LEER$
(256)                                          <UG>
1140 LU$="{CTL ,}{CTL ,}{CTL ,}{CTL ,}{C
TL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}{ESC CT
L >}{CTL ,}{CTL ,}":RO$="{CTL ,}{CTL ,}{
ESC CTL >}{CTL A}{CTL A}{CTL A}{CTL A}{C
TL A}{CTL A}{CTL A}{CTL ,}{CTL ,}"     <HJ>
1150 # MENUE_NEU                               <FA>
1160 POKE 82,%2:CLS                            <RJ>
1170 SETCOLOR %1,%0,12:SETCOLOR %2,%0,%0 <PA>
1180 ? "▴▴Mandelbrotmenge▴(Apfelmaennche
n)"                                            <VL>
1190 ? "nach▴Spektrum▴der▴Wissenschaft▴1
0/86"                                          <AP>
1200 ? "▴▴▴▴▴▴▴▴▴▴(c)▴by▴Stephan▴Gerle"       <BQ>
1210 POKE 82,5:?                               <CK>
1220 ? "Laden▴Bild"                            <XD>
1230 ? "Speichern▴Bild"                        <PA>
1240 ? "Berechnen▴Bild"                        <FW>
1250 ? "Ausschnitt▴vom▴Bild"                   <SZ>
1260 ? "Directory"                             <NA>
1270 ?                                         <WU>
1280 ? "Umschalten▴zum▴Bild:▴SELECT"           <MG>
1290 ? "Vom▴Bild▴zum▴Menue▴:▴START"            <FA>
1300 ?                                         <WB>
1310 ? "Aktuelle▴Parameter:"                   <QI>
1320 ? "-------------------"                   <LH>
1330 ? "Real-Teil▴▴▴▴▴▴▴(x)▴:▴";AECKE          <HO>
1340 ? "Imaginaer-Teil▴(y)▴:▴";BECKE           <OA>
1350 ? "Seitenlaenge▴▴▴(s)▴:▴";SEITE           <OA>
1360 ?                                         <WT>
1370 ? "Ihre▴Wahl?▴";                          <PR>
1380 MENUE=DPEEK(88)                           <FV>
1390 MENUEDLST=DPEEK(560)                      <GS>
1400 # START                                   <MT>
1410 DPOKE 560,MENUEDLST                       <HG>
1420 DPOKE 88,MENUE:POKE 87,%0                 <FN>
1430 POSITION 22,19:? "";                      <AM>
1440 REPEAT                                    <ET>
1450 UNTIL PEEK($D01F)=5 OR PEEK(764)<25
5                                              <CJ>
1460 IF PEEK($D01F)=5                          <DP>
1470 D=USR($0600,64)                           <HW>
1480 WHILE PEEK($D01F)<>6:WEND                 <JU>
1490 D=USR($0600,%0)                           <TK>
1500 ELSE                                      <TE>
1510 GET TT:TT=TT&95:T=%0                      <BQ>
1520 IF CHR$(TT)="S" THEN EXEC SPEICHERN <DP>
1530 IF CHR$(TT)="L" THEN EXEC LADEN     <BE>
1540 IF CHR$(TT)="B" THEN EXEC EINGABE:E
XEC RECHNEN:T=%1                               <EW>
1550 IF CHR$(TT)="A" THEN EXEC AUSSCHNIT
T:EXEC RECHNEN:T=%1                            <XW>
1560 IF CHR$(TT)="D" THEN ? :DIR :GET TT
:T=%1                                          <AO>
1570 IF T=%1 THEN POP :GO# MENUE_NEU           <FW>
1580 ENDIF                                     <TO>
1590 GO# START                                 <JH>
1600 --                                        <VQ>
1610 PROC EINGABE                              <YY>
1620 ?                                         <WN>
1630 ? "Realteil▴der▴linken▴unteren▴Ecke
"                                              <KC>
1640 INPUT "X=",AECKE                          <JF>
1650 ? "Imaginaerteil▴der▴linken▴unteren
▴Ecke"                                         <YG>
1660 INPUT "Y=",BECKE                          <KP>
1670 ? "Seitenlaenge▴des▴Ausschnitts"          <BR>
1680 INPUT "S=",SEITE                          <EG>
1690 ? "Maximale▴Iterationstiefe"              <RP>
1700 INPUT "T=",TIEFE                          <TQ>
1710 ENDPROC                                   <OT>
1720 --                                        <VY>
1730 PROC RECHNEN                              <NE>
1740 D=USR($0600,64)                           <HT>
1750 POKE AH,%0:MOVE AH,AH+%1,$1DFF            <SR>
1760 MOVE AH,AF,$1E00                          <CJ>
1770 SPALT=SEITE/160                           <VF>
1780 POKE 87,9:DPOKE 88,AH:COLOR 8             <CD>
1790 TEXT %0,%0,STR$(AECKE)                    <DC>
1800 TEXT %0,8,STR$(BECKE)                     <IL>
1810 TEXT %0,16,STR$(SEITE)                    <PP>
1820 DPOKE 88,AF:COLOR 6                       <VB>
1830 TEXT %0,%0,STR$(AECKE)                    <CM>
1840 TEXT %0,8,STR$(BECKE)                     <IX>
1850 TEXT %0,16,STR$(SEITE)                    <QB>
1860 FOR M=%0 TO 159                           <FY>
1870 BC=M*SPALT+BECKE                          <WQ>
1880 FOR N=%0 TO 159 STEP %2                   <ZV>
1890 AC=N*SPALT+AECKE                          <WI>
1900 AZ=%0:BZ=%0:ZAEHLER=%0                    <TW>
1910 REPEAT                                    <EU>
1920 AQ=AZ:A2=AZ^%2:B2=BZ^%2                   <HC>
1930 AZ=A2-B2+AC                               <PV>
1940 BZ=%2*AQ*BZ+BC                            <BF>
1950 ZAEHLER=ZAEHLER+%1                        <MD>
1960 UNTIL (A2+B2>4) OR (ZAEHLER>TIEFE)        <LO>
1970 IF ZAEHLER<=TIEFE                         <TB>
1980 COLOR ZAEHLER MOD 16                      <JQ>
1990 DPOKE 88,AH                               <BI>
2000 PLOT N DIV %2,191-M                       <WT>
2010 COLOR (ZAEHLER DIV 16) MOD 16             <AS>
2020 DPOKE 88,AF                               <YQ>
2030 PLOT N DIV %2,191-M                       <XC>
2040 ENDIF                                     <ST>
2050 IF PEEK($D01F)<>7                         <EG>
2060 IF PEEK($D01F)=6                          <ED>
2070 POKE 559,$22:D=USR($0600,64)              <YB>
2080 ELSE                                      <TT>
2090 IF PEEK($D01F)=5                          <DR>
2100 POKE 559,%0:D=USR($0600,%0)               <IX>
2110 ENDIF                                     <SM>
2120 ENDIF                                     <SP>
2130 ENDIF                                     <SS>
2140 NEXT N                                    <HH>
2150 NEXT M                                    <GZ>
2160 D=USR($0600,%0):GRAPHICS %0               <HX>
2170 ENDPROC                                   <PA>
2180 --                                        <WF>
2190 PROC SPEICHERN                            <TW>
2200 ?                                         <WA>
2210 ? "Nummer▴des▴Bildes▴";                   <QA>
2220 INPUT NR                                  <FV>
2230 FILE$="D:APFEL"                           <WV>
2240 FILE$(LEN(FILE$)+%1)=STR$(NR)             <CH>
2250 FILE$(LEN(FILE$)+%1)=".PIC"               <FK>
2260 TRAP #FEHLER                              <XI>
2270 OPEN #%1,8,%0,FILE$                       <CK>
```

Listing. »Apfel256«

```
2280 BPUT #%1,AH,$1E00                          <TY>
2290 BPUT #%1,AF,$1E00                          <SV>
2300 ? #%1,STR$(AECKE)                          <UI>
2310 ? #%1,STR$(BECKE)                          <VC>
2320 ? #%1,STR$(SEITE)                          <RQ>
2330 # FEHLER                                   <MC>
2340 CLOSE #%1                                  <HM>
2350 POP :GO# MENUE_NEU                         <NH>
2360 ENDPROC                                    <PB>
2370 --                                         <WG>
2380 PROC LADEN                                 <SN>
2390 ?                                          <XD>
2400 ? "Nummer des Bildes ";                    <QB>
2410 INPUT NR                                   <FW>
2420 FILE$="D:APFEL"                            <WW>
2430 FILE$(LEN(FILE$)+%1)=STR$(NR)              <CI>
2440 FILE$(LEN(FILE$)+%1)=".PIC"                <FL>
2450 TRAP #NICHT_VORHANDEN                      <NK>
2460 POKE 73,%0                                 <HD>
2470 OPEN #%1,4,%0,FILE$                        <AG>
2480 BGET #%1,AH,$1E00                          <MT>
2490 BGET #%1,AF,$1E00                          <LQ>
2500 INPUT #%1,FILE$                            <CU>
2510 AECKE=VAL(FILE$)                           <CO>
2520 INPUT #%1,FILE$                            <DA>
2530 BECKE=VAL(FILE$)                           <DA>
2540 INPUT #%1,FILE$                            <DG>
2550 SEITE=VAL(FILE$)                           <MD>
2560 # NICHT_VORHANDEN                          <XS>
2570 CLOSE #%1                                  <HZ>
2580 POP :GO# MENUE_NEU                         <NU>
2590 ENDPROC                                    <PO>
2600 --                                         <VR>
2610 PROC AUSSCHNITT                            <RB>
2620 D=USR($0600,65)                            <IF>
2630 PMB=112:POKE 54279,PMB                     <VA>
2640 PMB=PMB*256+1024+214                       <PF>
2650 POKE 559,58                                <CW>
2660 POKE 53256,%0:POKE 53257,%0                <WB>
2670 POKE 704,255:POKE 705,255                  <EP>
2680 POKE 53277,%2                              <FH>
2690 X=%0:Y=%0                                  <PK>
2700 AB=8                                       <SE>
2710 MOVE ADR(LEER$),PMB-214,256                <TJ>
2720 MOVE ADR(LEER$),PMB+256-214,256            <ZC>
2730 POKE 87,9:DPOKE 88,AH:COLOR 8              <BP>
2740 AA=AECKE:BB=BECKE:SPALT=SEITE/80           <BA>
2750 REPEAT                                     <FD>
2760 POKE AH,%0                                 <VH>
2770 MOVE AH,AH+%1,960                          <LV>
2780 TEXT %0,%0,STR$(AECKE)                     <DA>
2790 TEXT %0,8,STR$(BECKE)                      <JL>
2800 TEXT %0,16,STR$(SEITE)                     <PN>
2810 A=STICK(%0)                                <JJ>
2820 IF STRIG(%0)                               <XG>
2830 X=X+(A&8=%0)-(A&4=%0)                      <NR>
2840 Y=Y-(A&%2=%0)+(A&%1=%0)                    <PT>
2850 IF X<%0 THEN X=%0                          <QX>
2860 IF X+AB>80 THEN X=X-%1                     <IX>
2870 IF Y<%0 THEN Y=%0                          <SF>
2880 IF Y+AB>80 THEN Y=Y-%1                     <LG>
2890 ELSE                                       <UM>
2900 AB=AB+(A&8=%0)-(A&4=%0)                    <KW>
2910 IF AB<%1 THEN AB=%1                        <BH>
2920 IF X+AB>80 THEN AB=AB-%1                   <DD>
2930 IF Y+AB>80 THEN AB=AB-%1                   <DP>
2940 ENDIF                                      <TL>
2950 POKE 53248,X*%2+48                         <RS>
2960 POKE 53249,X*%2+40+AB*2                    <QZ>
2970 MOVE ADR(LU$),PMB-Y*%2,12                  <GD>
2980 MOVE ADR(RO$),PMB-Y*%2-%2*AB+264,12 <SP>
2990 AECKE=AA+X*SPALT                           <GR>
3000 BECKE=BB+Y*SPALT                           <GT>
3010 SEITE=AB*SPALT                             <LS>
3020 UNTIL PEEK($D01F)=6                        <YJ>
3030 POKE 53248,%0:POKE 53249,%0                <XB>
3040 D=USR($0600,%0):GRAPHICS %0                <HQ>
3050 ? "Maximale Iterationstiefe"               <QT>
3060 INPUT TIEFE                                <UL>
3070 ENDPROC                                    <OZ>
3080 --                                         <WE>
3090 PROC FARB256_INIT                          <SF>
3100 REM Grafik mit 256 Farben                  <UB>
3110 REM Aufruf mit D=USR($600,GTIAMode)        <TB>
3120 REM Ausschalten mit D=USR($600,0)          <YS>
3130 REM Helligkeitsbild bei AH=$8010           <FJ>
3140 REM Farbbild        bei AF=$A010           <CR>
3150 AH=$8010:AF=$A010                          <AS>
3160 POKE AH,%0:MOVE AH,AH+%1,$1DFF             <SK>
3170 MOVE AH,AF,$1E00                           <CC>
3180 POKE 106,120:REM Speicher schuetzen        <JX>
3190 --                                         <WJ>
3200 REM Displaylist erstellen                  <MW>
3210 --                                         <VN>
3220 A=$7800:ADA=AH:ADB=AF+40                   <ZL>
3230 POKE A,112:POKE A+%1,240:POKE A+%2,
240                                             <KF>
3240 FOR I=A+%3 TO A+%3+(191*%3) STEP 6         <UZ>
3250 POKE I,$CF                                 <KD>
3260 DPOKE I+%1,ADA                             <ES>
3270 ADA=ADA+80                                 <DG>
3280 POKE I+%3,$CF                              <XI>
3290 DPOKE I+4,ADB                              <CG>
3300 ADB=ADB+80                                 <DH>
3310 NEXT I                                     <FA>
3320 POKE I,65                                  <WY>
3330 DPOKE I+%1,A                               <NW>
3340 A=$7C00:ADA=AF:ADB=AH+40                   <EN>
3350 POKE A,112:POKE A+%1,112:POKE A+%2,
240                                             <IM>
3360 FOR I=A+%3 TO A+%3+(191*%3) STEP 6         <VH>
3370 POKE I,$CF                                 <KL>
3380 DPOKE I+%1,ADA                             <FA>
3390 ADA=ADA+80                                 <DO>
3400 POKE I+%3,$CF                              <WO>
3410 DPOKE I+4,ADB                              <BM>
3420 ADB=ADB+80                                 <DP>
3430 NEXT I                                     <FI>
3440 POKE I,65                                  <XG>
3450 DPOKE I+%1,A                               <OE>
3460 BLOAD "D:COL256.BIN":REM Grafikerwe
iterung laden                                   <EF>
3470 ENDPROC                                    <PH>
3480 --                                         <WM>
```

Listing. »Apfel256« (Schluß)

```
10000 PROC FARB256_INIT                         <VB>
10010 REM Grafik mit 256 Farben                 <EO>
10020 REM Aufruf mit D=USR($600,GTIAMode
)                                               <BF>
10030 REM Ausschalten mit D=USR($600,0)         <CX>
10040 REM Helligkeitsbild bei AH=$8010          <MN>
10050 REM Farbbild        bei AF=$A010          <PL>
10060 AH=$8010:AF=$A010                         <VI>
10070 POKE AH,%0:MOVE AH,AH+%1,$1DFF            <TU>
10080 MOVE AH,AF,$1E00                          <XM>
10090 POKE 106,120:REM Speicher schuetze
n                                               <ZO>
10100 --                                        <JF>
10110 REM Display-List aufbauen                 <VC>
10120 --                                        <JN>
10130 A=$7800:ADA=AH:ADB=AF+40                  <MP>
10140 POKE A,112:POKE A+%1,240:POKE A+%2
,240                                            <ZS>
10150 FOR I=A+%3 TO A+%3+(191*%3) STEP 6 <ZD>
10160 POKE I,$CF                                <SW>
10170 DPOKE I+%1,ADA                            <WA>
10180 ADA=ADA+80                                <LG>
10190 POKE I+%3,$CF                             <LF>
10200 DPOKE I+4,ADB                             <QZ>
10210 ADB=ADB+80                                <LD>
10220 NEXT I                                    <FX>
10230 POKE I,65                                 <DB>
10240 DPOKE I+%1,A                              <ZZ>
```

Listing. »FARB 256«



```
10250 A=$7C00:ADA=AF:ADB=AH+40                <SF>
10260 POKE A,112:POKE A+%1,112:POKE A+%2
,240                                          <YA>
10270 FOR I=A+%3 TO A+%3+(191*%3) STEP 6      <ZO>
10280 POKE I,$CF                              <TH>
10290 DPOKE I+%1,ADA                          <WL>
10300 ADA=ADA+80                              <KG>
10310 POKE I+%3,$CF                           <KF>
10320 DPOKE I+4,ADB                           <RK>
10330 ADB=ADB+80                              <LO>
10340 NEXT I                                  <GI>
10350 POKE I,65                               <DM>
10360 DPOKE I+%1,A:GRAPHICS 0                 <RG>
10370 BLOAD "D:COL256.BIN":REM Grafikerw
eiterung laden                                <IL>
10380 ENDPROC                                 <VM>
```

Listing. »FARB 256« (Schluß)

```
100 EXEC FARB256_INIT                         <LO>
110 D=USR($0600,64)                           <RT>
120 POKE 106,120                              <NE>
130 COL=3:LUM=6                               <OM>
140 DPOKE 88,AH                               <QN>
150 POKE 87,9                                 <OE>
160 FOR I=0 TO 79                             <AB>
170 COLOR INT(I/5)                            <HR>
180 PLOT I,0:DRAWTO I,191                     <MF>
190 NEXT I                                    <GJ>
200 DPOKE 88,AF                               <PC>
210 FOR I=0 TO 191                            <DY>
220 COLOR INT(I/12)                           <OX>
230 PLOT 0,I:DRAWTO 79,I                      <ZM>
240 NEXT I                                    <GA>
250 OPEN #1,4,0,"K:"                          <PL>
260 GET #1,J:CLOSE #1                         <RU>
270 D=USR($0600,0)                            <CT>
280 GRAPHICS 0:END                            <IH>
290 --                                        <JV>
300 PROC FARB256_INIT                         <PB>
310 AH=$8010:AF=$A010                         <HO>
320 POKE AH,%0:MOVE AH,AH+%1,$1DFF            <SM>
330 MOVE AH,AF,$1E00                          <IE>
340 POKE 106,120                              <NK>
350 A=$7800:ADA=AH:ADB=AF+40                  <OG>
360 POKE A,112:POKE A+%1,240:POKE A+%2,2
40                                            <WR>
370 FOR I=A+%3 TO A+%3+(191*%3) STEP 6        <SU>
380 POKE I,$CF                                <DJ>
390 DPOKE I+%1,ADA                            <PJ>
400 ADA=ADA+80                                <WM>
410 POKE I+%3,$CF                             <KR>
420 DPOKE I+4,ADB                             <NI>
430 ADB=ADB+80                                <XK>
440 NEXT I                                    <GC>
450 POKE I,65                                 <SU>
460 DPOKE I+%1,A                              <DS>
470 A=$7C00:ADA=AF:ADB=AH+40                  <SU>
480 POKE A,112:POKE A+%1,112:POKE A+%2,2
40                                            <UX>
490 FOR I=A+%3 TO A+%3+(191*%3) STEP 6        <SZ>
500 POKE I,$CF                                <CV>
510 DPOKE I+%1,ADA                            <OV>
520 ADA=ADA+80                                <WR>
530 POKE I+%3,$CF                             <KW>
540 DPOKE I+4,ADB                             <NN>
550 ADB=ADB+80                                <XP>
560 NEXT I                                    <GH>
570 POKE I,65                                 <SZ>
580 DPOKE I+%1,A:GRAPHICS 0                   <QN>
590 BLOAD "D:COL256.BIN"                      <LD>
600 ENDPROC                                   <KI>
```

Demonstrationsprogramm für 256 Farben

Zeichensätze selbstgestrickt

Benötigen Sie für Ihr Programm ein Copyright-Zeichen? Kein Problem. Mit unserem Programm lassen sich Zeichen einfach selbst entwerfen und in eigene Basic-Programme einbinden.

Stellen Sie sich folgende Szene vor: Sie sitzen an Ihrem Computer und tippen gerade die letzten Zeilen Ihres neuen Programms ein. Da durchfährt es Sie wie ein Blitz: Sämtliche Umlaute in den begleitendenden Bildschirmtexten haben Sie als »ae«, »oe« und »ue« eingegeben. Sie taten das ganz automatisch, denn standardmäßig hat der Atari Computer keine Umlaute auf der Tastatur, und im Laufe der Zeit schreibt man halt die Umlaute nach Kreuzworträtsel-Schema. Nur, schön sieht das natürlich nicht aus.

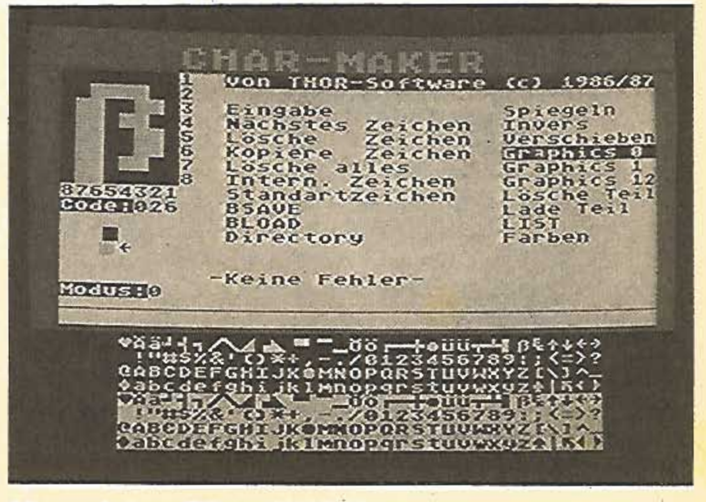
Der Atari hat zwar einen zweiten Zeichensatz eingebaut, und dieser enthält auch jede Menge verschiedener Sonderzeichen. Das deutsche »ß« ist jedoch nicht vorhanden.

Konstruieren lassen sich die Zeichen mit einem Zeichensatzprogramm. Und genau so etwas wollen wir Ihnen hier vorstellen: Ein Programm, mit dem Sie völlig problemlos eigene Zeichen malen und Zeichensätze in eigene Programme einbinden können.

Das Programm besteht aus zwei Teilen. Der erste Teil, das Hauptprogramm, ist in Turbo-Basic geschrieben. Dieses Programm benötigt ein weiteres in Maschinensprache geschriebenes Programm, das wir als AMPEL-Listing gedruckt haben. Speichern Sie es bitte auf der gleichen Diskette wie das Hauptprogramm.

Wenn Sie den »Char-Maker« starten, sehen Sie nach kurzer Zeit das Menü. In der linken oberen Ecke befindet sich das Fenster zum Malen der einzelnen Zeichen. Gleich darunter steht der Code des gerade bearbeiteten und im Fenster sichtbaren Zeichens. Unter diesem erscheint eine Anzeige, die Auskunft über die gerade eingeschaltete Farbe gibt, sowie eine Zeile, die über die verwendete Grafikstufe informiert.

Gleichzeitig leuchtet nach dem Starten des Programms in der oberen linken Ecke des Bildschirms ein Quadrat auf.



Alles auf einen Blick: »CHAR-MAKER«

```
Programmname :CHAR.OBJ
Länge        :2312 Bytes
----------------------------------

0000:FF FF 00 80 00 89 4C 49 <B9>
0008:83 4C 55 86 4C C7 86 4C <9C>
0010:4A 87 4C 73 87 4C DC 87 <17>
0018:4C 32 88 4C FA 83 4C 8C <52>
0020:88 4C A3 88 4C D1 88 4C <44>
0028:D1 88 4C D1 88 4C D1 88 <E8>
0030:4C D1 88 4C D1 88 4C D1 <66>
0038:88 00 00 00 00 00 00 00 <7C>
0040:00 00 00 00 00 00 00 00 <40>
0048:00 00 00 00 00 00 00 00 <48>
0050:00 80 01 80 02 00 03 80 <2E>
0058:03 F0 90 90 F0 90 A0 F0 <B3>
0060:47 80 80 42 00 00 02 02 <DD>
0068:02 02 02 02 02 02 02 02 <68>
0070:02 02 02 02 02 02 02 A0 <0E>
0078:20 42 A1 80 82 02 02 02 <BE>
0080:82 02 02 41 59 80 00 00 <5B>
0088:00 00 23 28 21 32 0D 2D <80>
0090:21 2B 25 32 00 00 00 00 <B3>
0098:00 00 00 00 00 00 0A 00 <AC>
00A0:34 00 0A 00 00 00 00 00 <FB>
00A8:00 00 00 00 00 00 00 00 <A8>
00B0:00 00 00 00 00 00 00 00 <B0>
00B8:00 00 00 00 00 00 00 00 <B8>
00C0:00 00 00 00 00 00 00 00 <C0>
00C8:00 00 00 00 00 00 00 00 <C8>
00D0:00 00 00 00 00 00 00 00 <D0>
00D8:00 00 00 00 00 00 00 00 <D8>
00E0:00 00 00 00 00 00 00 00 <E0>
00E8:00 00 00 00 00 00 00 00 <E8>
00F0:00 00 00 00 00 00 00 00 <F0>
00F8:00 00 00 00 00 00 00 00 <F8>
0100:00 00 00 00 00 00 00 00 <02>
0108:00 00 00 00 00 00 00 00 <0A>
0110:00 00 00 00 00 00 00 00 <12>
0118:00 00 00 00 00 00 00 00 <1A>
0120:00 00 00 00 00 00 00 00 <22>
0128:00 00 00 00 00 00 00 00 <2A>
0130:00 00 00 00 00 00 00 00 <32>
0138:00 00 00 00 00 00 00 00 <3A>
0140:00 00 00 00 00 00 00 00 <42>
0148:00 00 00 00 00 00 00 00 <4A>
0150:00 00 00 00 00 00 00 00 <52>
0158:00 00 00 00 00 00 00 00 <5A>
0160:00 00 00 00 00 00 00 00 <62>
0168:00 00 00 00 00 00 00 00 <6A>
0170:00 00 00 00 00 00 00 00 <72>
0178:00 00 00 00 00 00 00 00 <7A>
0180:00 00 00 00 00 00 00 00 <82>
0188:00 00 00 00 00 00 00 00 <8A>
0190:00 00 00 00 00 00 00 00 <92>
0198:00 00 00 00 00 00 00 00 <9A>
01A0:00 00 00 00 00 00 00 8A <2C>
01A8:0A A8 B9 4B 80 85 CB B9 <C6>
01B0:4C 80 18 6D 57 80 85 CC <54>
01B8:60 AE 78 02 A0 00 8A 29 <A0>
01C0:01 D0 01 88 8A 29 02 D0 <DD>
01C8:01 C8 8C 3A 80 A0 00 8A <8F>
01D0:29 04 D0 01 88 8A 29 08 <37>
01D8:D0 01 C8 98 AA AC 3A 80 <84>
01E0:60 AD 33 80 0A 0A 18 69 <5D>
01E8:30 AA AD 34 80 0A 0A 18 <56>
01F0:69 10 A8 60 8E 3C 80 8C <0E>
01F8:3D 80 20 DB 81 EC 3C 80 <9F>
0200:D0 05 A9 00 8D 49 80 CC <1D>
0208:3D 80 D0 0C A9 00 8D 4A <EF>
0210:80 EC 3C 80 D0 02 18 60 <F5>
0218:38 60 AD 47 80 18 6D 49 <A2>
0220:80 8D 47 80 AD 48 80 18 <1E>
0228:6D 4A 80 AE 37 80 F0 04 <DF>
0230:18 6D 4A 80 8D 48 80 A2 <F5>
0238:00 A0 02 86 CB 98 18 6D <1E>
0240:57 80 85 CC A0 7F A2 03 <6C>
0248:A9 00 91 CB 88 C0 FF D0 <98>
0250:F9 AC 48 80 BD 53 80 91 <AE>
0258:CB C8 CA 10 F7 AE 47 80 <AB>
0260:8E 00 D0 60 4C 3C 83 A5 <2A>
0268:14 48 29 10 D0 04 68 49 <16>
0270:FF 48 68 29 8F 8D 12 D0 <00>
0278:AD 35 80 D0 E7 AD 78 02 <1C>
0280:C9 0F F0 0A CD A0 80 D0 <81>
0288:05 EE 95 80 D0 0B A9 00 <31>
0290:8D 95 80 AD 78 02 8D A0 <F0>
0298:80 AD 95 80 C9 40 90 07 <20>
02A0:A9 40 8D 95 80 D0 05 A5 <32>
02A8:14 4A 90 B8 AD 38 80 F0 <C9>
02B0:1A AD 47 80 29 04 F0 13 <6A>
02B8:AD 34 80 C9 08 B0 0C AD <94>
02C0:33 80 C9 09 B0 05 A9 0B <1E>
02C8:8D 78 02 AE 47 80 AC 48 <35>
02D0:80 20 EE 81 B0 69 20 B3 <7C>
02D8:81 8E 3C 80 8C 3D 80 A9 <01>
02E0:00 8D 94 80 AD 38 80 F0 <CD>
02E8:11 AD 33 80 C9 08 B0 0A <DE>
02F0:AD 34 80 C9 08 B0 03 EE <CB>
02F8:94 80 18 AD 33 80 6D 3C <63>
0300:80 AC 94 80 F0 05 6D 3C <94>
0308:80 29 FE C9 28 90 08 A9 <1D>
0310:00 8D 3C 80 AD 33 80 8D <AF>
0318:33 80 8C 08 D0 AD 34 80 <A7>
0320:18 6D 3D 80 C9 12 90 08 <B2>
0328:A9 00 8D 3D 80 AD 34 80 <AB>
0330:8D 34 80 AD 3D 80 8D 4A <F1>
0338:80 AD 3C 80 8D 49 80 20 <EB>
0340:14 82 A9 79 8D 00 02 A9 <AF>
0348:84 8D 01 02 4C 62 E4 68 <CF>
0350:EE 35 80 A9 40 8D 0E D4 <39>
0358:AD 57 80 8D 07 D4 A9 03 <00>
0360:8D 1D D0 AD 2F 02 09 0C <58>
0368:8D 2F 02 A9 01 8D 49 80 <67>
0370:8D 4A 80 A9 01 8D 6F 02 <C6>
0378:A9 FF 8D 0C D0 A9 07 A2 <B6>
0380:82 A0 61 20 5C E4 A9 01 <DC>
0388:8D 49 80 8D 4A 80 A5 58 <72>
0390:8D 5E 80 A5 59 8D 5F 80 <72>
0398:A9 59 8D 30 02 A9 80 8D <7E>
03A0:31 02 A9 79 8D 00 02 A9 <7D>
03A8:84 8D 01 02 A9 C0 8D 0E <C1>
03B0:D4 A2 00 EE 35 80 8A 8E <2C>
03B8:94 80 20 26 84 AE 94 80 <89>
03C0:9D A1 80 E8 D0 F0 20 D5 <3F>
03C8:84 A9 00 8D 35 80 60 18 <46>
03D0:A5 58 65 CD 8D 42 80 A5 <B4>
03D8:59 69 00 8D 43 80 A5 CE <6B>
03E0:F0 13 AD 42 80 18 69 28 <D2>
03E8:8D 42 80 90 03 EE 43 80 <74>
03F0:C6 CE 18 90 E9 AD 42 80 <5B>
03F8:85 CD AD 43 80 85 CE 60 <88>
0400:68 EE 35 80 68 68 85 CD <09>
0408:68 68 85 CE 68 68 8D 3A <01>
0410:80 20 C9 83 A0 00 AD 3A <5A>
0418:80 F0 0D B1 CD 49 80 91 <2A>
0420:CD C8 CE 3A 80 18 90 EE <69>
0428:CE 35 80 60 A2 00 8E 46 <F0>
0430:80 C9 00 08 29 7F C9 20 <24>
0438:B0 04 69 40 90 07 C9 60 <BC>
0440:B0 03 38 E9 20 28 10 05 <39>
0448:09 80 EE 46 80 48 29 7F <01>
0450:A2 00 86 CC 0A 26 CC 0A <41>
0458:26 CC 0A 26 CC 85 CB A5 <B9>
0460:CC 6D 58 80 85 CC 68 60 <40>
0468:8E 0A D4 CA D0 FA 60 68 <67>
0470:D8 18 69 01 8D 00 02 68 <78>
0478:69 00 8D 01 02 68 40 48 <4D>
0480:8A 48 A2 0F 20 62 84 A2 <38>
0488:04 BD 96 80 9D 16 D0 CA <43>
```

Das ist der Cursor des Programms. Mit ihm werden die einzelnen Menüpunkte ausgewählt. Wenn Sie jetzt einen Joystick an Port 1 anschließen, können Sie den Cursor mit dem Joystick über den gesamten Bildschirm bewegen. Fahren Sie einmal über einzelne Menüpunkte hinweg. Sie sehen dann, daß die Menüpunkte, über denen der Cursor gerade steht, invers dargestellt werden. Indem Sie den Joystickknopf drücken, wählen Sie den jeweiligen Menüpunkt an. Die einzelnen Menüpunkte bedeuten folgendes:

Eingabe: Damit läßt sich ein Zeichen aus dem Zeichensatz in den Zeichenbildschirm kopieren, um es zu verändern. Nachdem Sie diesen Menüpunkt angewählt haben, erscheint unter der Codezahl ein kleiner Pfeil. Indem sie den Joystick dann hinauf- oder hinunterdrücken, läßt sich die Codezahl abwandeln. Der Pfeil ist auch nach rechts und links verschiebbar, um die Zehner- und Hunderterstellen der Codezahl zu verändern. Durch Druck auf den Joystickknopf beenden Sie die Eingabe, und das ausgewählte Zeichen erscheint im Zeichenfenster.

Nächstes Zeichen: Mit diesem Menüpunkt wählen Sie das Zeichen mit der darauffolgenden Codezahl aus und holen es vom Zeichensatz ins Zeichenfenster.

Lösche Zeichen: Das Zeichen mit der ausgewählten Codezahl wird gelöscht.

Kopiere Zeichen: Kopiert ein über die Codezahl auswählbares Zeichen an die gerade bearbeitete Stelle des Zeichensatzes.

Lösche alles: Löscht den gesamten Zeichensatz.

Intern. Zeichensatz: Hiermit kopiert man den Atari-Zeichensatz mit den internationalen Sonderzeichen in den Arbeitsspeicher des Programms.

Standardzeichen: Wie oben, nur mit dem normalem Zeichensatz.

BSAVE: Speichert einen Zeichensatz auf Diskette.

BLOAD: Lädt einen Zeichensatz von Diskette.

Directory: Zeigt das Inhaltsverzeichnis der Diskette an. Gleichzeitig läßt sich ein Zeichensatz wie bei BLOAD laden, indem der jeweilige Directory-Eintrag mit dem Cursor ausgewählt wird.

Spiegeln: Das Zeichen im Zeichenfenster wird gespiegelt, wobei zwei Spiegelachsen zur Verfügung stehen. Diese werden durch Hoch-Nieder- oder Links-Rechts-Bewegung mit dem Joystick ausgewählt.

Invers invertiert das Zeichen im Zeichenfenster. Helle Punkte werden in dunkle und dunkle Punkte in helle Punkte umgewandelt.

Verschieben: Verschiebt ein Zeichen in der angegebenen Richtung. Die Richtung wird durch den Joystick ausgewählt. Bildschirmpunkte, die aus dem Zeichenfenster hinausgeschoben werden, kommen am gegenüberliegenden Rand wieder zum Vorschein.

Graphics 0 stellt den gesamten Zeichensatz in dieser Grafikstufe dar. Ein Zeichen besteht aus acht mal acht Punkten.

Graphics 1 stellt den Zeichensatz in Grafikstufe 1 dar. Ein Zeichen besteht ebenfalls aus acht mal acht Punkten. Eine extra Anzeigemöglichkeit für Grafikstufe 2 ist nicht eingebaut, da die Zeichen dort zwar doppelt so hoch dargestellt werden, ansonsten aber genauso aufgebaut sind wie in Grafikstufe 1.

Graphics 12: In einem Zeichen lassen sich jetzt vier Farben darstellen.

```
0490:10 F7 68 AA 20 69 84 48 <D9>
0498:8A 48 AE 2F 02 CA 8E 00 <E6>
04A0:D4 AD 58 80 09 02 8D 09 <53>
04A8:D4 8D 0A D4 A2 04 BD 9B <AE>
04B0:80 9D 16 D0 CA 10 F7 68 <D4>
04B8:AA 20 69 84 48 AD 58 80 <0A>
04C0:8D 0A D4 EA 8D 09 D4 20 <02>
04C8:69 84 48 AD 58 80 09 02 <D2>
04D0:8D 0A D4 EA 8D 09 D4 20 <33>
04D8:69 84 40 A9 11 8D 6F 02 <66>
04E0:A9 FF 8D 09 D0 8D 0A D0 <B7>
04E8:8D 0B D0 8D 0C D0 A9 30 <EF>
04F0:8D 01 D0 8D 02 D0 8D 03 <AD>
04F8:D0 8D 07 D0 A9 38 8D 06 <B6>
0500:D0 A9 40 8D 05 D0 A9 48 <5E>
0508:8D 04 D0 EE 35 80 A2 00 <A3>
0510:20 A1 81 20 8C 85 A2 01 <1B>
0518:20 A1 81 20 8C 85 A2 02 <24>
0520:20 A1 81 20 8C 85 A2 03 <4D>
0528:20 A1 81 20 8C 85 A2 00 <92>
0530:20 A1 81 A9 FF A0 10 91 <88>
0538:CB C8 C0 30 D0 F9 A9 10 <58>
0540:8D 94 80 AD 38 80 F0 05 <15>
0548:A9 40 8D 94 80 A0 3C A9 <25>
0550:50 2D 94 80 20 96 85 A2 <D6>
0558:01 20 A1 81 A0 40 A9 40 <B6>
0560:2D 94 80 20 96 85 A2 02 <C1>
0568:20 A1 81 A0 44 A9 40 2D <06>
0570:94 80 20 96 85 A2 03 20 <9A>
0578:A1 81 A0 40 A9 10 2D 94 <1F>
0580:80 20 96 85 A9 40 A0 48 <A9>
0588:2D 94 80 20 96 85 CE 35 <AB>
0590:80 60 A0 7F A9 00 91 CB <12>
0598:88 10 FB 60 A2 03 91 CB <4D>
05A0:C8 CA 10 FA 60 AE 38 80 <4C>
05A8:D0 16 8E 3B 80 8E 3C 80 <D2>
05B0:AE 46 80 F0 02 49 FF 8D <F2>
05B8:3D 80 49 FF 8D 3A 80 60 <0E>
05C0:A2 03 A0 00 8C 3A 80 8C <28>
05C8:3B 80 8C 3C 80 8C 3D 80 <AC>
05D0:48 29 C0 48 49 C0 D0 06 <74>
05D8:20 19 86 18 90 1B 68 48 <71>
05E0:29 80 F0 06 20 2A 86 18 <8A>
05E8:90 0F 68 48 29 40 F0 06 <18>
05F0:20 3B 86 18 90 03 20 4C <A0>
05F8:86 68 E0 00 F0 1F 0E 3A <AD>
0600:80 0E 3A 80 0E 3B 80 0E <87>
0608:3B 80 0E 3C 80 0E 3C 80 <43>
0610:0E 3D 80 0E 3D 80 68 0A <09>
0618:0A 48 CA 10 B4 68 60 AD <1E>
0620:3D 80 09 03 8D 3D 80 AD <41>
0628:3A 80 29 FC 8D 3A 80 60 <02>
0630:AD 3C 80 09 03 8D 3C 80 <85>
0638:AD 3A 80 29 FC 8D 3A 80 <D0>
0640:60 AD 3B 80 09 03 8D 3B <AF>
0648:80 AD 3A 80 29 FC 8D 3A <3B>
0650:80 60 AD 3A 80 09 03 8D <20>
0658:3A 80 60 68 EE 35 80 68 <4E>
0660:68 20 26 84 A5 CB 85 CD <4B>
0668:A5 CC 85 CE A0 00 8C 3F <BC>
0670:80 A2 10 8E 40 80 AC 3F <21>
0678:80 B1 CD 20 9F 85 A2 00 <1B>
0680:20 A1 81 AC 40 80 AD 3A <4F>
0688:80 20 96 85 A2 01 20 A1 <CA>
0690:81 AC 40 80 AD 3B 80 20 <D3>
0698:96 85 A2 02 20 A1 81 AC <97>
06A0:40 80 AD 3C 80 20 96 85 <74>
06A8:A2 03 20 A1 81 AC 40 80 <0B>
06B0:AD 3D 80 20 96 85 AD 40 <D0>
06B8:80 18 69 04 8D 40 80 EE <4A>
06C0:3F 80 AD 3F 80 C9 08 D0 <D5>
06C8:AD CE 35 80 60 68 EE 35 <DA>
06D0:80 68 68 8D 3A 80 68 68 <85>
06D8:8D 3B 80 68 68 8D 3C 80 <CC>
06E0:68 68 8D 3D 80 AD 3A 80 <BF>
06E8:20 26 84 AC 3C 80 B1 CB <65>
06F0:A8 AE 38 80 D0 0C AE 3B <B2>
06F8:80 AD 3D 80 20 28 87 18 <1E>
0700:90 15 AD 3D 80 29 01 AE <55>
0708:3B 80 20 28 87 A8 E8 AD <71>
0710:3D 80 29 02 20 28 87 20 <90>
0718:44 87 CE 35 80 60 01 02 <B4>
0720:04 08 10 20 40 80 FE FD <2B>
0728:FB F7 EF DF BF 7F 8D 40 <86>
0730:80 98 3D 20 87 8D 41 80 <76>
0738:AD 40 80 F0 09 AD 41 80 <BD>
0740:1D 18 87 8D 41 80 AD 41 <4F>
0748:80 60 AC 3C 80 91 CB 60 <9B>
0750:EE 35 80 68 68 68 AA 09 <7C>
0758:40 A8 8A 09 80 8C 73 80 <8B>
0760:8D 76 80 8E 77 80 8E 79 <3A>
0768:80 8D 7A 80 8E 7B 80 8E <71>
0770:7C 80 8E 78 80 CE 35 80 <62>
0778:60 68 EE 35 80 68 68 20 <26>
0780:26 84 A0 07 8C 3A 80 B1 <FA>
0788:CB 20 92 87 AC 3A 80 91 <E5>
0790:CB 88 10 F0 CE 35 80 60 <CB>
0798:AE 38 80 D0 06 20 A1 87 <51>
07A0:18 90 03 20 B0 87 60 A2 <F3>
07A8:07 8D 3B 80 A9 00 4E 3B <C7>
07B0:80 2A CA 10 F9 60 A2 03 <0F>
07B8:8D 3B 80 A9 00 8D 3C 80 <81>
07C0:AD 3B 80 29 C0 0D 3C 80 <A1>
07C8:8D 3C 80 E0 00 F0 0F 0E <24>
07D0:3B 80 0E 3B 80 4E 3C 80 <F7>
07D8:4E 3C 80 CA 10 E2 AD 3C <C6>
07E0:80 60 68 EE 35 80 68 68 <56>
07E8:20 26 84 A0 07 B1 CB AE <C9>
07F0:38 80 D0 05 49 FF 18 90 <0B>
07F8:03 20 FF 87 91 CB 88 10 <AA>
0800:EC CE 35 80 60 A2 03 8D <64>
0808:3A 80 A9 00 8D 3B 80 AD <2F>
0810:3A 80 18 D8 69 40 2A 2A <4E>
0818:2A 29 03 0D 3B 80 8D 3B <A6>
0820:80 E0 00 F0 0F 0E 3B 80 <0A>
0828:0E 3B 80 0E 3A 80 0E 3A <E0>
0830:80 CA 10 DB AD 3B 80 60 <19>
0838:68 EE 35 80 68 68 8D 3A <C6>
0840:80 68 68 8D 3B 80 AD 3A <6F>
0848:80 20 26 84 68 68 8D 3C <61>
0850:80 AD 3C 80 F0 11 A0 07 <E6>
0858:B1 CB 20 65 88 91 CB 88 <AA>
0860:10 F6 CE 3C 80 D0 EA CE <48>
0868:35 80 60 AE 3B 80 D0 0C <1D>
0870:20 82 88 AE 38 80 F0 03 <AF>
0878:20 82 88 60 20 86 88 AE <F8>
0880:38 80 F0 03 20 86 88 60 <40>
0888:0A 69 00 60 4A 90 02 09 <97>
0890:80 60 68 EE 35 80 68 68 <A5>
0898:20 26 84 85 D1 A5 CB 85 <20>
08A0:D4 A5 CC 85 D5 CE 35 80 <71>
08A8:60 68 EE 35 80 68 68 85 <C9>
08B0:CC A9 00 85 CB 85 CD AD <E0>
08B8:58 80 85 CE A9 FF 8D 01 <A5>
08C0:D3 A2 04 A0 00 B1 CB 91 <1A>
08C8:CD 88 D0 F9 E6 CC E6 CE <C2>
08D0:CA D0 F2 CE 35 80 60 68 <E5>
08D8:60 00 00 00 00 00 00 00 <98>
08E0:00 00 00 00 00 00 00 00 <F0>
08E8:00 00 00 00 00 00 00 00 <F8>
08F0:00 00 00 00 00 00 00 00 <00>
08F8:00 00 00 00 00 00 00 00 <08>
0900:00 00 00 00 00 00 00 00 <12>
```

Listing. »CHAR.OBJ« wird vom Char-Maker benötigt. Bitte mit AMPEL eingeben.

Lösche Teil: Damit läßt sich ein beliebig langer Block von Zeichen aus dem Zeichensatz löschen. Mit der Codezahl werden das erste und das letzte zu löschende Zeichen ausgewählt.

Lade Teil: Aus einem auf der Diskette vorhandenem Zeichensatz wird ein beliebig langer Teil in den gerade bearbeiteten Zeichensatz geladen. Das erste und das letzte zu ladende Zeichen wird über die Codezahl ausgewählt.

LIST: Mit diesem Befehl läßt sich ein beliebiger Teil aus einem Zeichensatz in ein Basic-Programm einbauen. Das Programm fragt dann, an welcher Speicherseite es stehen soll (siehe auch Beitrag »Die Handschrift des Computers« in diesem Heft). Dann wird nach dem ersten und letzten zu übertragenden Zeichen gefragt. Wollen Sie den gesamten Zeichensatz übertragen, so geben Sie 0 und 127 ein. Anschließend folgt die Frage, ob ein zweiter Block mit Zeichen gespeichert werden soll. Diese beantwortet man mit »ja«, indem man den Joystickknopf drückt, oder mit »nein«, indem man den Joystick hinauf oder hinunter bewegt. Dann folgt die Frage nach der ersten Zeilennummer im Programm sowie nach der Schrittweite der Nummern. Zuletzt fragt das Programm nach dem Namen, den es dem File geben soll. Auf der Diskette wird dann ein ASCII-File unter dem angegebenen Namen gespeichert. Dieses wird mit dem »ENTER«-Befehl in ein Basic-Programm eingebunden. Das auf der Diskette erzeugte File enthält gleichzeitig ein Ladeprogramm, das den Zeichensatz unter Basic installiert.

Farben verändert die gerade ausgewählte Farbe.

Ein Zeichen editiert man, indem man mit dem Cursor ins Zeichenfenster fährt und dort durch Druck auf den Joystickknopf Punkte setzt oder löscht. Ein Zeichen läßt sich zusätzlich auswählen, indem Sie mit dem Cursor auf die Codezahl fahren und den Knopf drücken. Sie erhalten dann die gleiche Funktion wie beim Menüpunkt »Eingabe«.

Tritt beim Laden oder Speichern auf Diskette ein Fehler auf, so wird die Fehlernummer unter den Menüpunkten gezeigt. Um mit dem Programm fortzufahren, müssen Sie dann den Joystickknopf drücken. (Thomas Richter/hf)

Steckbrief	
Programm:	Char-Maker
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer und AMPEL
Datenträger:	Diskette

```
2 DPOKE 566,DPEEK(528)                          <OY>
5 TRAP 5:CLOSE #1:OPEN #1,4,0,"D:CHAR3.O
BJ":CLOSE #1                                    <DH>
10 GRAPHICS 0:POKE 559,0:DLO=DPEEK(560):
PAUSE 0:COL=$809B                               <WB>
15 POKE COL,0:POKE COL+1,10:POKE COL+2,0
:POKE COL+4,0                                   <BJ>
20 BLOAD "D:CHAR3.OBJ":POKE 559,34              <YG>
30 POKE $8058,$A0:POKE $A000,0:MOVE $A00
```

Listing. »CHAR.BAS«, bitte mit Prüfsummer eingeben



```
0,$A001,$03FF                                  <AN>
40 U=USR($8000):MCO=3:GRAF=0:AS=32:POKE
705,$32:POKE 711,0:POKE 706,$76:POKE 707
,$1A                                           <YS>
50 POKE 756,$CC:SETCOLOR 0,8,6:SETCOLOR
4,8,0:DIM F$(20),FI$(20),H$(40),AN(16),E
N(16),ANAD(16),ENAD(16)                        <EY>
60 EXEC MENUE:YA=-1:EXEC GR0                   <SM>
65 --                                          <ZL>
70 X=PEEK($8033):Y=PEEK($8034):XH=11+18*
(X>28):POKE 77,0                               <OV>
80 IF X<8 AND Y<8 THEN GO# ZEEID               <EA>
81 IF Y=9 AND X>=5 AND X<=7 THEN X=11:Y=
2:GOTO 120                                     <GQ>
90 IF (Y<2 OR Y>11) OR (X<11 OR (X>26 AN
D X<29)) THEN IF YA>0:U=USR($8015,XA,YA,
11+5*(XA<29)):YA=-1:ENDIF :GOTO 110            <BD>
100 IF Y<>YA AND YA>-1 THEN U=USR($8015,
XH,YA,11+5*(XA<29)):U=USR($8015,XH,Y,11+
5*(XH<29)):XA=XH:YA=Y                          <DY>
101 IF Y<>YA AND YA=-1 THEN U=USR($8015,
XH,Y,11+5*(XH<29)):YA=Y:XA=XH                  <VL>
110 IF PEEK($D01F)=5 AND STICK(0)=15 THE
N EXEC FARBUM                                  <HK>
120 IF STRIG(0)=1 THEN 70                      <BU>
122 IF (X>=1 AND X<=4) AND (Y>=11 AND Y<
=14) THEN MCO=Y-11:EXEC GONG:IF GRAF<12:
MCO=3*MCO:ENDIF :EXEC PARAM:GOTO 70            <OK>
125 IF (Y<2 OR Y>11) OR (X<11 OR (X>26 A
ND X<29)) THEN EXEC GONG:GOTO 70               <DW>
130 MN=(Y-2)+10*(X>28)+1:PAUSE 0:EXEC GO
NG                                             <TK>
135 IF STICK(0)<15 THEN 70                     <AL>
140 ON MN EXEC EIN,NAE,LOE,KOP,ALO,INE,S
TA,BSA,BLO,DIR,SPI,INV,VER,GR0,GR1,GR2,D
RE,PRO,LIS,END                                 <YW>
150 POKE 77,0:PAUSE 5:GOTO 70                  <QD>
151 --                                         <JP>
152 # NEWPO:U=USR($8015,XA,YA,11+5*(XA<2
9)):XHN=11+18*(XN>28):U=USR($8015,XHN,YN
,11+5*(XHN<29)):GOTO 70                        <XK>
155 --                                         <KB>
160 # ZEEID:XA=-1:YA=-1                        <TK>
170 PAUSE 1:X=PEEK($8033):Y=PEEK($8034)        <JH>
180 IF X>7 OR Y>7 THEN YA=-1:GOTO 70           <TU>
185 IF STICK(0)=15 AND PEEK($D01F)=5 THE
N EXEC FARBUM                                  <NV>
190 IF STRIG(0)=1 THEN 170                     <ZN>
195 BW=MCO:IF GRAF<>12 AND MCO=3 THEN BW
=1                                             <JU>
196 IF GRAF=12 THEN U=USR($8006,AS,(7-X)
-1,Y,BW):GOTO 210                              <QJ>
200 U=USR($8006,AS,7-X,Y,BW)                   <AZ>
210 U=USR($8003,AS):POKE $D01F,0:XA=X:YA
=Y                                             <SK>
220 GOTO 170                                   <OK>
999 --                                         <LD>
1000 PROC MENUE:POKE 752,1:POKE 82,8:POK
E 83,39:POKE 87,0:? "{ESC CTL <}";:POSIT
ION 8,0                                        <AZ>
1010 ? "1▲▲von▲THOR-Software▲(c)▲1986/87
";                                             <MC>
1020 ? "2"                                     <KX>
1030 ? "3▲▲Eingabe▲▲▲▲▲▲▲▲▲▲▲Spiegeln"         <MX>
1040 ? "4▲▲N{CTL K}chstes▲Zeichen▲▲Inver
s"                                             <SE>
1050 ? "5▲▲L{CTL O}sche▲▲▲Zeichen▲▲Versc
hieben";                                       <KB>
1060 ? "6▲▲Kopiere▲▲Zeichen▲▲Graphics▲0"       <DM>
1070 ? "7▲▲L{CTL O}sche▲alles▲▲▲▲▲▲Graph
ics▲1"                                         <JX>
1080 ? "8▲▲Intern.▲Zeichen▲▲▲Graphics▲12
"                                              <FE>
1090 POSITION 0,8:? "87654321▲▲▲Standart
zeichen▲▲▲L{CTL O}sche▲Teil";                  <WI>
1100 ? "▲▲▲BSAVE▲▲▲▲▲▲▲▲▲▲▲▲▲Lade▲Teil"        <BZ>
1110 ? "▲▲▲BLOAD▲▲▲▲▲▲▲▲▲▲▲▲▲LIST"             <PS>
1120 ? "▲▲▲Directory▲▲▲▲▲▲▲▲▲Farben"           <MI>
1121 POSITION 0,16:? "___________________
_____________________";                        <MJ>
1130 EXEC PARAM:ENDPROC                        <TQ>
1140 PROC PARAM:FOR I=11 TO 14:POSITION
2,I:? "▲▲▲▲";:NEXT I                           <BN>
1150 POSITION 0,9:? "Code:";:VL=AS:EXEC
PRIN                                           <DK>
1160 POSITION 0,15:? "Modus:";GRAF;"▲▲";       <HW>
1170 POSITION 10,14:? "-Keine▲Fehler-"         <JK>
1180 IF GRAF=0 OR GRAF=1                       <KF>
1190 POSITION 4,11+(MCO=3):? "{ESC ESC}{
ESC CTL +}";                                   <OZ>
1200 ELSE :POSITION 2,11+MCO:? "{ESC ESC
}{ESC CTL +}";:ENDIF                           <KC>
1210 ENDPROC                                   <OJ>
1220 PROC PRIN:H$="000"                        <SJ>
1230 F$=STR$(VL):H$(4-LEN(F$))=F$:? H$;:
ENDPROC                                        <BT>
1240 PROC FARBUM:IF GRAF=0 OR GRAF=1           <WF>
1250 POSITION 4,11+(MCO=3):? "▲";              <GB>
1260 ELSE :POSITION 2,11+MCO:? "▲";:ENDI
F                                              <JC>
1270 MCO=MCO+1+2*(GRAF<>12)                    <EP>
1280 IF MCO>3 THEN MCO=0                       <UF>
1290 EXEC PARAM:EXEC GONG:ENDPROC              <PY>
1300 PROC GONG:FOR I=15 TO 0 STEP -1:SOU
ND 0,120,10,I:SOUND 1,121,10,I:PAUSE 0:N
EXT I:ENDPROC                                  <GX>
1310 PROC PLOF:POKE $8035,1:POKE 53248,0
:ENDPROC                                       <CR>
1320 PROC PLON:POKE $8035,0:ENDPROC            <MW>
1330 PROC SHURE                                <IK>
1340 SOUND 0,60,10,10:REPEAT :UNTIL STRI
G(0)=1                                         <SE>
1350 POSITION 11,17:? "Ganz▲sicher?(Feue
rknopf!)";:YE=0:PAUSE 3                        <WP>
1360 FOR I=10 TO 0 STEP -0.2                   <QY>
1370 SOUND 0,144,10,I:SOUND 1,145,10,I         <KS>
1380 IF STRIG(0)=0 THEN YE=1:EXIT              <PE>
1390 IF STICK(0)<>15 THEN EXIT                 <JG>
1400 PAUSE 0:NEXT I:SOUND 0,0,0,0:SOUND
1,0,0,0                                        <HK>
1405 REPEAT :UNTIL STRIG(0)=1                  <WV>
1410 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:IF YE=0 THEN
 POP :POP :GOTO 70                             <MJ>
1420 ENDPROC                                   <OQ>
1430 PROC DISKIN:DPOKE 560,DLO:POKE $D40
E,$40:SETCOLOR 4,3,4:SETCOLOR 2,0,10:SET
COLOR 1,0,0                                    <FA>
1450 POKE 82,0:POSITION 0,19                   <TS>
1470 ? "▲▲▲▲▲Diskettenzugriff!▲Bitte▲▲▲"       <SG>
1480 ? "▲▲▲▲▲einen▲Augenblick▲Geduld▲▲!"       <GZ>
1500 POKE 82,8:POSITION 8,2:EXEC PLOF          <BC>
1510 ENDPROC                                   <OP>
1520 PROC DISKOU:U=USR($8000):EXEC PLON:
SETCOLOR 4,8,0:SETCOLOR 0,8,6                  <EE>
1521 POSITION 0,18:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT DEL}
{ESC SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT
 DEL}{ESC SHIFT DEL}{ESC SHIFT DEL}";:U=
USR($8003,AS):ENDPROC                          <XO>
1530 PROC FILEIN:POKE 752,0                    <BJ>
1540 TRAP 1540:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";          <WC>
1550 POSITION 11,17:? "Filename▲D1:";          <QF>
1560 INPUT "{ESC CTL *}{ESC CTL +}",F$:F
I$="D:":FI$(3)=F$:POSITION 0,17:? "{ESC
SHIFT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";
:TRAP 40000:POKE 752,1:? "{ESC CTL *}{ES
C CTL +}";:ENDPROC                             <LH>
1570 PROC BEGENI:ASS=AS                        <FR>
1580 POSITION 11,17:? "Erstes▲▲Zeichen"        <CH>
1590 EXEC EIN:AN=AS                            <SP>
1600 POSITION 11,17:? "Letztes▲Zeichen"        <UY>
1610 EXEC EIN:EN=AS:AS=ASS:POSITION 5,9:
VL=AS:EXEC PRIN                                <YT>
1620 ANAD=USR($8018,AN):ANI=PEEK(209)          <AG>
1630 ENAD=USR($8018,EN):ENI=PEEK(209)          <IJ>
1640 POSITION 11,17:? "{ESC SHIFT >}{ESC
 SHIFT DEL}{ESC SHIFT DEL}";:ENDPROC           <ME>
2000 PROC EIN:EXEC PLOF                        <QS>
2010 H$=STR$(AS):X=7:FI$="000":FI$(4-LEN
```




```
(H$))=H$:PAUSE 2                            <CU>
2020 PAUSE 2:ST=STICK(0):W=ASC(FI$(X-4))
-48:SOUND 0,0,0,0                           <NZ>
2030 IF ST=11 AND X>5 THEN X=X-1:W=ASC(F
I$(X-4))-48                                 <IJ>
2040 IF ST=7 AND X<7 THEN X=X+1:W=ASC(FI
$(X-4))-48                                  <AZ>
2050 IF ST=13 THEN W=W-1:IF W=-1 THEN W=
9                                           <RJ>
2060 IF ST=14 THEN W=W+1:IF W=10 THEN W=
0                                           <FO>
2070 FI$(X-4,X-4)=CHR$(W+48):POSITION 5,
9:? FI$                                     <XR>
2080 POSITION X-1,10:? "▲{ESC ESC}{ESC C
TL -}▲";:IF ST<>15 THEN SOUND 0,60,10,10 <FG>
2090 IF STRIG(0)=1 THEN 2020                <DG>
2091 AS=VAL(FI$):IF AS>127 THEN EXEC GON
G:GOTO 2020                                 <QU>
2100 U=USR($8003,AS):VL=AS:POSITION 5,10
:? "▲▲▲▲";:SOUND 0,0,0,0:EXEC PLON:ENDPR
OC                                          <DR>
2110 PROC NAE:AS=AS+1:IF AS=128 THEN AS=
0                                           <NK>
2120 U=USR($8003,AS):POSITION 5,9:VL=AS:
EXEC PRIN:ENDPROC                           <WY>
2130 PROC LOE:EXEC SHURE                    <SV>
2140 U=USR($8018,AS)                        <VU>
2150 FOR I=U TO U+7:POKE I,0:NEXT I:U=US
R($8003,AS):ENDPROC                         <MA>
2160 PROC KOP:POSITION 11,17:? "Welches▲
Zeichen▲?"                                  <BQ>
2170 ASS=AS:EXEC EIN                        <VL>
2180 U=USR($8018,ASS):V=USR($8018,AS)       <DU>
2190 MOVE V,U,8:AS=ASS:U=USR($8003,AS):P
OSITION 0,17:? "{ESC SHIFT >}{ESC SHIFT
DEL}{ESC SHIFT DEL}";:POSITION 5,9:VL=AS
:EXEC PRIN:ENDPROC                          <TM>
2200 PROC ALO:EXEC SHURE:POKE $A000,0:MO
VE $A000,$A001,$03FF                        <NG>
2210 U=USR($8003,AS):ENDPROC                <FR>
2220 PROC INE:EXEC SHURE:U=USR($801B,$CC
):U=USR($8003,AS):ENDPROC                   <VL>
2230 PROC STA:EXEC SHURE:U=USR($801B,$E0
):U=USR($8003,AS):ENDPROC                   <AI>
2240 PROC BSA:EXEC FILEIN:POKE 752,0        <MQ>
2250 TRAP 2250:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";       <VO>
2260 POSITION 11,17:? "In▲welche▲Page?▲9
6{ESC CTL +}{ESC CTL +}";                   <NG>
2270 INPUT "{ESC CTL *}{ESC CTL +}",PG:P
OKE 752,1                                   <EN>
2280 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";                 <ES>
2290 EXEC DISKIN                            <SS>
2300 TRAP #ERRO:OPEN #2,8,0,FI$             <FO>
2310 PUT #2,$FF,$FF,0,PG,$FF,PG+3           <YU>
2320 BPUT #2,$A000,$0400                    <XU>
2330 CLOSE #2:EXEC DISKOU:TRAP 40000:END
PROC                                        <IJ>
2340 PROC BLO:EXEC FILEIN                   <QW>
2350 EXEC DISKIN:TRAP #ERRO                 <DO>
2360 OPEN #2,4,0,FI$:GET #2,A,A             <YJ>
2370 IF A+A<>510 THEN 2400                  <QC>
2380 GET #2,A,A,A,A                         <DX>
2390 BGET #2,$A000,$0400                    <RG>
2400 CLOSE #2:TRAP 40000:EXEC DISKOU:U=U
SR($8003,AS):ENDPROC                        <AN>
2500 # ERRO:TRAP #ERRO:CLOSE #2:EXEC DIS
KOU                                         <TL>
2510 SOUND 0,12,12,15:SOUND 1,15,12,15      <OX>
2520 PAUSE 40:SOUND 0,0,0,0:SOUND 1,0,0,
0:POSITION 10,14:? "▲▲Error▲-▲";ERR;"▲▲▲ <QP>
2530 REPEAT :UNTIL STRIG(0)=0               <VK>
2540 POSITION 10,14:? "▲▲▲▲-Keine▲Fehler
-";:POP :POP :GOTO 70                       <TP>
2550 PROC DIR:EXEC DISKIN                   <BU>
2560 TRAP #ERRO:OPEN #2,6,0,"D:*.*"         <NO>
2570 TRAP 2620:C=0:POKE 82,8:? "{ESC CTL
*}{ESC CTL +}";:BI=DPEEK(88)                <HH>
2580 FOR I=0 TO 11:AD=BI+I*40+8:POKE AD,
0:MOVE AD,AD+1,31:NEXT I:FOR I=690 TO 69
3:POKE I,255:NEXT I                         <UB>
2585 POKE 93,0:E=1                          <HA>
2590 TRAP 2620:INPUT #2,F$:FI$="":IF F$(
2,2)<>"▲" THEN E=0                          <YP>
2591 TRAP 2592:INPUT #2,FI$                 <AH>
2592 IF F$(2,2)="▲" THEN F$=F$(3)           <JU>
2593 IF FI$<>"" THEN IF FI$(2,2)="▲":FI$
=FI$(3):ELSE :POSITION 8,7:? "{ESC SHIFT
DEL}":E=0:ENDIF                             <YZ>
2600 POSITION 8,C:? F$;"▲";FI$;:C=C+1:IF
C>=7 AND E THEN EXEC WTFK:C=0:GOTO 2580 <VO>
2610 GOTO 2590                              <UK>
2620 TRAP 40000:EXEC WTFK:EXEC DISKOU:EX
EC MENUE:CLOSE #2:YA=-1:ENDPROC             <CS>
2630 PROC WTFK:EXEC DISKOU:POSITION 2,17
:? "Bitte▲ausw{CTL K}hlen▲oder▲START▲dr{
CTL J}cken!":EXEC DISKCHOI                  <OX>
2640 REPEAT :UNTIL PEEK($D01F)=7:POSITIO
N 2,17:? "{ESC SHIFT >}{ESC SHIFT DEL}{E
SC SHIFT DEL}{ESC SHIFT >}";:EXEC DISKIN
:ENDPROC                                    <AP>
2650 PROC SPI:POSITION 11,17:EXEC PLOF      <EG>
2660 ? "Spiegeln:▲{ESC ESC}{ESC CTL -}{E
SC ESC}{ESC CTL =}":SP=0                    <CJ>
2670 ST=STICK(0)                            <JI>
2680 IF ST=13 OR ST=14 THEN SP=0:POSITIO
N 21,17:? "{ESC ESC}{ESC CTL -}{ESC ESC}
{ESC CTL =}":EXEC GONG                      <FD>
2690 IF ST=7 OR ST=11 THEN SP=1:POSITION
21,17:? "{ESC ESC}{ESC CTL +}{ESC ESC}{
ESC CTL *}":EXEC GONG                       <XK>
2700 IF STRIG(0)=1 THEN 2670                <NY>
2710 IF SP=1 THEN 2750                      <NM>
2720 U=USR($8018,AS):MOVE U,$0680,8:FOR
I=0 TO 7:POKE U+I,PEEK($0680+(7-I)):NEXT
I                                           <SZ>
2730 U=USR($8003,AS):GOTO 2760              <JO>
2750 U=USR($800C,AS):U=USR($8003,AS)        <DB>
2760 EXEC PLON:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";:EN
DPROC                                       <RK>
2770 PROC INV:U=USR($800F,AS):U=USR($800
3,AS):ENDPROC                               <XC>
2780 PROC VER:SP=0:EXEC PLOF                <VH>
2790 POSITION 11,17:? "Verschieben:▲{ESC
ESC}{ESC CTL -}"                            <AE>
2800 ST=STICK(0)                            <IR>
2810 IF ST=14 THEN SP=0:POSITION 24,17:?
"{ESC ESC}{ESC CTL -}":EXEC GONG            <VU>
2811 IF ST=13 THEN SP=1:POSITION 24,17:?
"{ESC ESC}{ESC CTL =}":EXEC GONG            <YA>
2820 IF ST=7 THEN SP=2:POSITION 24,17:?
"{ESC ESC}{ESC CTL *}":EXEC GONG            <KQ>
2821 IF ST=11 THEN SP=3:POSITION 24,17:?
"{ESC ESC}{ESC CTL +}":EXEC GONG            <AP>
2830 IF STRIG(0)=1 THEN 2800                <JC>
2840 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";                 <ES>
2850 POSITION 11,17:? "Wie▲oft▲?▲1":OFT=
1                                           <VG>
2860 PAUSE 3:ST=STICK(0):SOUND 0,0,0,0      <UI>
2870 IF ST=13 AND OFT>1 THEN OFT=OFT-1      <JU>
2880 IF ST=14 AND OFT<7 THEN OFT=OFT+1      <LE>
2890 IF ST<>15 THEN SOUND 0,60,10,10        <LO>
2900 POSITION 21,17:? CHR$(OFT+48);         <NW>
2910 IF STRIG(0)=1 THEN 2860                <PE>
2920 ON SP+1 EXEC UP,DO,LR,LR               <WR>
2930 U=USR($8003,AS):EXEC PLON:POSITION
0,17:? "{ESC SHIFT >}{ESC SHIFT DEL}{ESC
SHIFT DEL}";:ENDPROC                        <EU>
2940 PROC UP:U=USR($8018,AS):FOR I=1 TO
OFT:A=PEEK(U):MOVE U+1,U,7:POKE U+7,A:NE
XT I:ENDPROC                                <ZD>
2950 PROC DO:U=USR($8018,AS):FOR I=1 TO
OFT                                         <RO>
2960 A=PEEK(U+7):-MOVE U,U+1,7:POKE U,A:
NEXT I:ENDPROC                              <HA>
```

Listing. »CHAR.BAS« (Fortsetzung)

```
2970 PROC LR:SW=1                                 <AA>
2980 IF SP=3 THEN SW=0                            <SO>
2990 U=USR($8012,AS,SW,OFT):ENDPROC               <WV>
3000 PROC GR0                                     <IZ>
3010 U=USR($8009,2):POKE $8038,0:GRAF=0:
POKE COL+1,PEEK(707):POKE COL+2,PEEK(711
):POKE COL+4,PEEK(711)                            <UH>
3015 EXEC CLFAU:ENDPROC                           <SJ>
3020 PROC GR1                                     <JS>
3030 U=USR($8009,6):POKE $8038,0:GRAF=1           <WS>
3040 FOR I=COL TO COL+3:POKE I,PEEK(707)
:NEXT I:POKE COL+4,PEEK(711):EXEC CLFAU:
ENDPROC                                           <TU>
3050 PROC GR2                                     <KO>
3060 U=USR($8009,4):POKE $8038,1:GRAF=12          <VV>
3070 FOR I=0 TO 2:POKE COL+I,PEEK(705+I)
:NEXT I:POKE COL+3,$C8:POKE COL+4,PEEK(7
11):EXEC CLFAU                                    <WK>
3071 POKE $8049,255:ENDPROC                       <MC>
3080 PROC CLFAU:U=USR($8000)                      <DS>
3090 FOR I=11 TO 14                               <RR>
3100 IF I<13 THEN POSITION 4,I:? "▲";             <TM>
3110 POSITION 2,I:? "▲";                          <AX>
3120 NEXT I:IF (MCO=1 OR MCO=2) AND GRAF
<>12 THEN MCO=3                                   <MQ>
3130 EXEC PARAM:PAUSE 0:U=USR($8003,AS):
PAUSE 0:ENDPROC                                   <CP>
3140 PROC DRE                                     <ST>
3150 EXEC SHURE:EXEC BEGENI                       <AR>
3160 POKE ANAD,0:MOVE ANAD,ANAD+1,(ENAD-
ANAD)-1+7:U=USR($8003,AS):ENDPROC                 <OA>
3170 PROC PRO:EXEC BEGENI:EXEC FILEIN             <BB>
3180 EXEC DISKIN:TRAP #ERRO                       <DU>
3190 OPEN #2,4,0,FI$                              <LG>
3200 GET #2,A,B:IF A+B<>510 THEN 3300             <UE>
3210 GET #2,A,B,C,D                               <HD>
3220 BGET #2,$A800,$0400                          <VI>
3230 MOVE ANAD+$0800,ANAD,ENAD-ANAD+7             <TF>
3300 CLOSE #2:EXEC DISKOU                         <FT>
3310 TRAP 40000:U=USR($8003,AS):ENDPROC           <YD>
3320 PROC LIS                                     <YX>
3330 POKE 752,0:POSITION 0,17:? "{ESC SH
IFT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";:P
OSITION 11,17:? "Ab▲welcher▲Page?▲96{ESC
CTL +}{ESC CTL +}";                               <HJ>
3340 TRAP 3330:INPUT "{ESC CTL *}{ESC CT
L +}",BEG                                         <HI>
3350 BEAD=BEG*256:POKE 752,1:? "{ESC CTL
*}{ESC CTL +}";:K=2                               <RW>
3360 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:TRAP 40000            <DM>
3365 IF PEEK($D01F)=6 THEN ANAD(K-2)=32:
ENAD(K-2)=127:GOTO 3410                           <VA>
3370 EXEC BEGENI:ANAD(K-2)=ANAD:ENAD(K-2
)=ENAD:AN(K-2)=ANI:EN(K-2)=ENI:IF ENAD<A
NAD THEN 3370                                     <HC>
3380 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:POSITION 0,1
7:? K;"ten▲Block▲eingeben?(Knopf=ja,{ESC
ESC}{ESC CTL -}{ESC ESC}{ESC CTL =}=nei
n)";:EXEC PLOF                                    <XM>
3390 IF STRIG(0)=0 AND K<18 THEN K=K+1:G
OTO 3360                                          <PJ>
3400 IF STICK(0)=15 AND K<18 THEN 3390            <QV>
3410 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:TRAP 3410:PO
KE 752,0                                          <AR>
3420 POSITION 11,17:? "Startzeilennummer
:";:INPUT "{ESC CTL *}{ESC CTL +}",ZN             <ON>
3430 TRAP 3430:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";             <VW>
3440 POSITION 11,17:? "Schrittweite:10{E
SC CTL +}{ESC CTL +}";                            <HU>
3450 INPUT "{ESC CTL *}{ESC CTL +}",ZS:T
RAP 40000                                         <TI>
3460 EXEC FILEIN                                  <MR>
3470 EXEC DISKIN:TRAP #ERRO                       <DX>
3480 OPEN #2,8,0,FI$                              <NN>
3490 ? #2;ZN;"BA=";BEAD;":RESTORE";ZN:ZO
=ZN:EXEC INC                                      <XT>
3500 ? #2;ZN;"READ▲A:IF▲A<0▲THEN▲";ZO+ZS
*3:EXEC INC                                       <ZO>
3510 ? #2;ZN;"POKE▲AD,A:AD=AD+1:GOTO▲";Z
O+ZS:EXEC INC                                     <AG>
3520 ? #2;ZN;"IF▲A=-1000▲THEN▲RETURN":EX
EC INC                                            <SD>
3530 ? #2;ZN;"AD=BA+8*INT(ABS(A)):GOTO";
ZO+ZS:EXEC INC                                    <UA>
3540 FOR J=2 TO K                                 <OF>
3550 ? #2;ZN;"DATA▲-";AN(J-2);".1":EXEC
INC                                               <RO>
3560 AD=ANAD(J-2):FOR L=AN(J-2) TO EN(J-
2)                                                <AU>
3570 ? #2;ZN;"DATA";:EXEC INC                     <IW>
3580 FOR I=0 TO 7                                 <AH>
3590 ? #2;PEEK(AD);:AD=AD+1                       <PP>
3600 IF I<7 THEN ? #2;",";                        <RG>
3610 IF I=7 THEN ? #2                             <NL>
3620 NEXT I:NEXT L:NEXT J                         <PW>
3625 ? #2;ZN;"DATA▲-1000"                         <IV>
3630 CLOSE #2:EXEC DISKOU:TRAP 40000:END
PROC                                              <IQ>
3640 PROC INC:ZN=ZN+ZS:ENDPROC                    <ZS>
3650 PROC END:POSITION 11,17                      <QG>
3660 ? "Helligkeit:{ESC ESC}{ESC CTL -}{
ESC ESC}{ESC CTL =}▲Farbe:{ESC ESC}{ESC
CTL +}{ESC ESC}{ESC CTL *}"                       <MY>
3670 IF MCO=0 THEN REG=COL-2                      <BH>
3680 IF MCO>0 AND GRAF=12 THEN REG=704+M
CO                                                <XM>
3690 IF MCO>0 AND GRAF<>12 THEN REG=707           <OZ>
3700 EXEC PLOF                                    <HT>
3710 W=PEEK(REG)                                  <TM>
3720 PAUSE 2:ST=STICK(0)                          <DG>
3730 IF ST&1=0 AND (W MOD 16)>0 THEN W=W
-1                                                <SL>
3740 IF ST&2=0 AND (W MOD 16)<15 THEN W=
W+1                                               <CR>
3750 IF ST&4=0 THEN W=W-16:IF W<0 THEN W
=W+256                                            <PS>
3760 IF ST&8=0 THEN W=W+16:IF W>255 THEN
 W=W-256                                          <SQ>
3770 POKE REG,W                                   <PA>
3780 IF STRIG(0)=1 THEN 3720                      <LR>
3790 IF REG=COL-2 THEN POKE 711,W                 <AA>
3800 IF GRAF=12 THEN EXEC GR2                     <HL>
3810 IF GRAF=0 THEN EXEC GR0                      <TP>
3820 IF GRAF=1 THEN EXEC GR1                      <VI>
3830 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:EXEC PLON:EN
DPROC                                             <SW>
3840 PROC DISKCHOI:YA=-1:BI=DPEEK(88)             <XB>
3850 IF PEEK($D01F)=6 THEN ENDPROC                <DD>
3855 IF STRIG(0)=0 THEN IF H=0 AND STICK
(0)=15 AND NOT (X<8 OR Y>6 OR (X>18 AND
X<24) OR X>34) THEN 3920                          <ZW>
3860 X=PEEK($8033):Y=PEEK($8034)                  <AE>
3865 H=0:XH=8+16*(X>23):W=(PEEK(XH+40*Y+
BI))&127:IF W<33 OR W>59 THEN H=1                 <TT>
3870 IF X<8 OR Y>6 OR (X>18 AND X<24) OR
 H OR X>34 THEN IF YA>=0:U=USR($8015,XA,
YA,11):YA=-1:ENDIF :GOTO 3850                     <YS>
3880 IF XA=XH AND Y=YA THEN 3850                  <TF>
3890 IF YA>=0 THEN U=USR($8015,XA,YA,11)          <JW>
3900 XA=XH:YA=Y:U=USR($8015,XA,YA,11)             <SM>
3910 IF STRIG(0)=1 THEN 3850                      <PD>
3920 FI$="▲▲▲▲▲▲▲▲▲▲▲"                            <OD>
3930 BD=BI+XH+Y*40                                <BK>
3940 MOVE BD,ADR(FI$),11                          <YE>
3950 BD=ADR(FI$):FOR I=BD TO BD+11                <ZY>
3960 W=PEEK(I):W=(W&127)+32:POKE I,W              <FL>
3970 NEXT I                                       <GE>
3980 FOR I=1 TO 8                                 <BS>
3990 IF FI$(I,I)="▲" THEN EXIT                    <UT>
4000 NEXT I                                       <ES>
4010 F$(1,I-1)=FI$:F$(I,I)=".":F$(I+1)=F
I$(9,11):CLOSE #2                                 <TN>
4020 EXEC GONG:FI$="D:":FI$(3)=F$:POP :P
OP :EXEC MENUE:YA=-1:GOTO 2350                    <FC>
```

Listing. »CHAR.BAS« (Schluß)

Ein toller Malkasten

Viele Firmen bringen Malprogramme auf den Markt, die sich nur in wenigen Details voneinander unterscheiden. Unser Malprogramm »Grafik-Draw« vereint viele Funktionen und besticht mit seinen vielfältigen Grafikstufen.

Malprogramme für die Atari 8-Bit-Computer sind genügend im Angebot. Von Datasoft gibt es den »Micropainter«, Atari und Koala-Soft haben das »Atari-Artist«-Programm beziehungsweise den »Koala-Painter« herausgebracht, und ein weiteres amerikanisches Programm namens »Fun with Art« schaltet auf Wunsch in jeder Zeile über einen Display-List-Interrupt die Farben um. Ganz zu schweigen von den vielen anderen Malprogrammen kleinerer Softwarefirmen. Warum sollte man also ein weiteres derartiges Programm veröffentlichen?

Zum einen haben die professionellen Programme den Nachteil, daß sie nur für viel Geld zu bekommen sind. Zum anderen unterstützen sie nicht die vielfältigen Grafikstufen der XL/XE-Computer. Grund genug, ein Programm zu veröffentlichen, das die meisten Grafikstufen berücksichtigt, eine umfangreiche Bibliothek an Malfunktionen besitzt, sowie obendrein noch eine Hardcopy-Routine aufweist, mit der ein Bild bequem und schnell auf einem CP-80-Drucker von Seikosha ausgedruckt wird. Wenn Sie nur eine Zeile im Programm ändern, arbeitet es auch mit anderen Druckern zusammen.

Das Programm »Grafik-Draw« besteht insgesamt aus drei Listings. Listing 1 ist das eigentliche Malprogramm. Programmteile, die zeitkritische Aufgaben übernehmen (Listing 2 und 3), sind in Maschinensprache programmiert und werden vom Hauptprogramm nach dessen Start von der Diskette dazugeladen. Diese Listings müssen mit dem AMPEL-Programm eingegeben und unter den Namen »UPADSTK.BIN« und »UKIPDRK.BIN« auf der gleichen Diskette wie das Hauptprogramm gespeichert werden.

Nach dem Start von Grafik-Draw erscheint auf dem Bildschirm das Hauptmenü. Der leuchtende Balken wählt die einzelnen Menüpunkte aus. Er wird mit einem Joystick oder einem Grafik-Tablett wie der Atari-Maltafel bewegt, die in Port 1 anzuschließen sind. Um zwischen dem Joystick oder der Maltafel als Eingabemedium zu entscheiden, geht man auf den Menüpunkt »Zeichengerät« und schaltet dann auf Knopfdruck zwischen den beiden Eingabegeräten hin und her. Das gerade gültige Zeichengerät wird rechts neben dem Menüpunkt angezeigt.

Das Programm stellt elf Zeichen sowie sechs weitere Funktionen zur Verfügung. Wurde eine Zeichenfunktion angewählt, so schaltet der Computer vom Menü zum gezeichneten Bild. Dort wird mit der Taste <F> die Malfarbe festgelegt. Zurück ins Menü kommt man mit der <SPACE>-Taste. Folgende Zeichenfunktionen stehen zur Auswahl:

Zeichnen: Freihändiges Zeichnen wird mit dieser Funktion unterstützt. Die Linie auf dem Bildschirm folgt direkt dem Fadenkreuz.

Linie: Beim ersten Druck auf den Joystick/Touch-Tablet-Knopf wird der Anfangspunkt, mit dem zweiten Knopfdruck der Endpunkt einer zu ziehenden Linie festgelegt.

Linie folgend: Wie »Linie«, nur daß der Endpunkt einer ersten Linie zugleich der Anfangspunkt einer zweiten Linie ist.

Kreis: Der erste Knopfdruck legt den Mittelpunkt eines zu zeichnenden Kreises fest. Durch Bewegen des Fadenkreuzes bestimmt man den Durchmesser des Kreises.

Ellipse: Wie Kreis, nur daß hierbei eine Ellipse gezeichnet wird, die in der Horizontalen und Vertikalen unterschiedliche Ausdehnungen hat.

Rechteck: Der erste Knopfdruck bestimmt den obersten linken Eckpunkt eines Rechtecks, der zweite Knopfdruck den untersten rechten Eckpunkt. Durch diese beiden Punkte wird ein Rechteck gezeichnet.

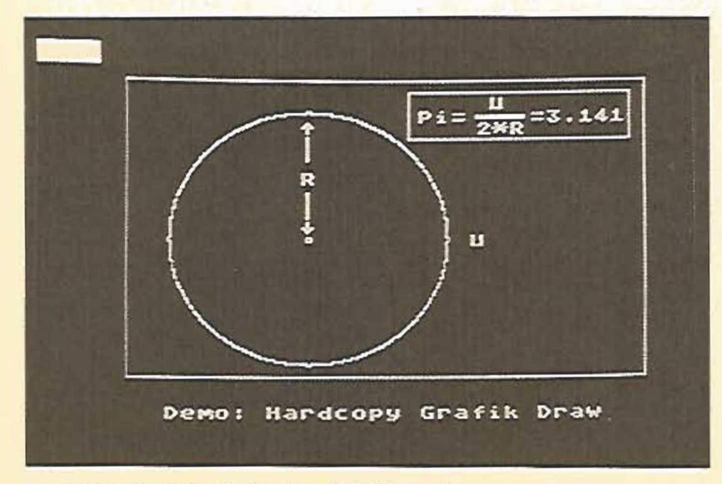
Rechteck gef.: Wie »Rechteck«. Anschließend wird das Rechteck ausgefüllt.

Ausfüllen: Malt eine beliebige schwarze Fläche wird in der gewählten Farbe aus.

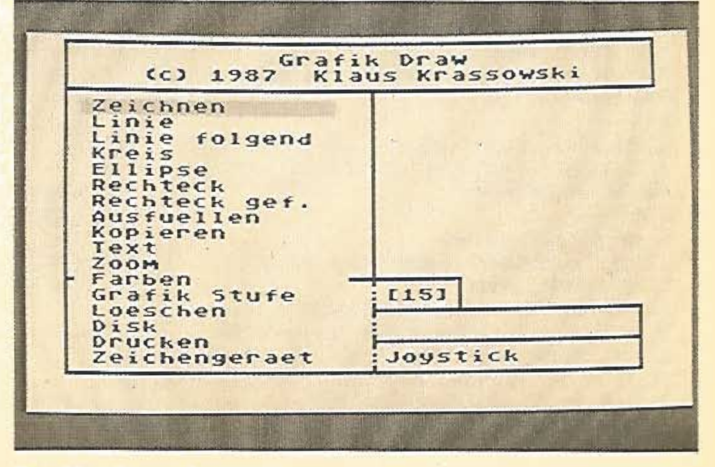
Kopieren: Diese Funktion kopiert einen beliebig großen rechteckigen Bildschirmausschnitt in einen anderen Teil des Bildes. Läßt man den Joystick/Maltafelknopf während der Bewegung des Ausschnitts gedrückt, wird mit dem Ausschnitt gezeichnet.

Text: Maximal 20 Zeichen lassen sich eingeben. Mit dem Joystick positioniert man dann den Text auf dem Bildschirm. Läßt man den Joystickknopf gedrückt, wird mit dem Text wie bei »Kopieren« auf Bildschirm gezeichnet.

Zoom: Mit dem Fadenkreuz wird ein rechteckiger Bildschirmausschnitt bestimmt, der über den ganzen Bildschirm vergrößert wird. In der Vergrößerung läßt sich nach Belieben zeichnen.



Beispiel einer technischen Zeichnung



Das Menü von Grafik-Draw

Die folgenden sechs Menüpunkte sprechen die restlichen Funktionen des Programms an:

Farben: Auf dem Bildschirm erscheinen vier Farbbalken. Neben dem obersten Balken befindet sich ein Pfeil, der die gerade gewählte Farbe markiert. Mit <START> wählt man eine andere Farbe an. <SELECT> verändert die Helligkeit der gewählten Farbe und <OPTION> variiert den Farbton. Durch <ESC> erfolgt die Übernahme der Veränderungen und die Rückkehr in den Zeichenmodus.

Grafik Stufe stellt die Auflösung des Bildes ein. Die einzelnen Stufen entsprechen denen in Atari-Basic. Zur Auswahl stehen Stufe 5, 7, 8, 9, 10, 11 und 15. Die gewählte Stufe wird rechts neben dem Menüpunkt angezeigt. Stufe 5 bietet 80 mal 48 Punkte, wobei vier Farben frei zu definieren sind. In Stufe 7 erhöht sich die Auflösung auf 160 mal 96 Bildschirmpunkte. Die höchste Auflösung bietet Stufe 8 mit 320 mal 192 Punkten. Hierbei steht aber nur noch eine Farbe in zwei Helligkeiten zur Verfügung. Stufe 9, 10 und 11 sind Unterformen von Stufe 8. Hier werden jeweils vier nebeneinanderliegende Punkte zu einem zusammengefaßt. Die Auflösung beträgt in diesen Stufen 80 mal 192. In Stufe 9 sind 16 Helligkeiten in einer Farbe verfügbar, und in Stufe 11 stehen alle 16 Farbschattierungen einer Helligkeitsstufe zur Auswahl. In Stufe 10 sind neun Farben wählbar, wobei jeder eine von insgesamt 256 Farben zugeordnet wird. In dieser so gut wie überhaupt nicht genutzten Grafikstufe des Ataris lassen sich sehr ansprechende Bilder zeichnen.

Löschen: Nach einer Sicherheitsabfrage wird das Bild gelöscht.

Disk: Das Programm verzweigt ins Disketten-Untermenü.
- <D> zeigt das Directory von Laufwerk 1 an.
- <L> lädt ein Bild von der Diskette. Dabei wird je nach Bild automatisch auf die richtige Grafikstufe umgeschaltet und die richtigen Farben gewählt. Bilder vom Atari-Artist Programm lassen sich nur dann laden, wenn sie mit <INSERT> gespeichert wurden. In diesem Fall legt der Atari-Artist die Bilder nicht in komprimierter Form auf Diskette ab.
- <S> speichert ein Bild auf Diskette.
- <ESC> kehrt ins Menü zurück.

Drucken erzeugt eine Hardcopy auf einem CP-80-Drucker. Dabei stehen zwei Vergrößerungsstufen zur Auswahl.
- <1> produziert eine Hardcopy über die halbe Breite einer DIN-A4-Seite.
- <2> druckt ein Bild über eine volle DIN-A4-Seite aus.

Bei einer Hardcopy über die halbe Breite des Blattes werden Sie zusätzlich gefragt, ob Sie den Ausdruck in der Mitte des Bildes oder nach links oder rechts versetzt haben wollen. Der Ausdruck paßt sich dann entsprechend dieser Eingabe an. Auf diese Weise lassen sich zum Beispiel doppelseitige Einladungen drucken.

Mit diesem Menüpunkt läßt sich ein Bild normal oder invers aufs Blatt nadeln. Farben werden automatisch in Grauwerte umgewandelt. Der Menüpunkt »Drucken« funktioniert nicht mit den Grafikstufen 5 und 7.

Falls Sie einen anderen Drucker als den CP-80 von Seikosha besitzen, läßt sich die Hardcopy-Routine folgendermaßen anpassen: In Zeile 10230 des Turbo-Basic-Programms steht der Drucker-Steuercode zum Drucken von Bitmustern. In diese Zeile müssen Sie die Steuercodes Ihres eigenen Druckers eintragen. Für einen Epson kompatiblen Drucker muß die Zeile wie folgt aussehen:

```
10230 GR$=CHR$(27):GR$(%2)="K":GR$(%3)=CHR$(192):
GR$(4)=CHR$(0)
```

Das Programm belegt fast den ganzen Speicher des Computers. Weitere umfangreiche Funktionen lassen sich also nicht integrieren.

Wir hoffen, daß Sie mit Grafik-Draw jetzt nicht mehr ständig viele verschiedene Malprogramme zur Hand haben müssen, um in verschiedenen Grafikstufen zeichnen zu können. Viel Spaß mit dem Programm!

(Klaus Krassowski/hf)

Steckbrief	
Programm:	Grafik-Draw
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer und AMPEL
Datenträger:	Diskette

```
10 --                                          <YW>
11 REM                                         <JQ>
12 REM Graphic Draw                            <AM>
13 REM (c) 1987  Klaus Krassowski              <BE>
15 REM                                         <JY>
49 --                                          <ZR>
500 GOSUB 10000:GOSUB 8000                     <DK>
600 BLOAD "D:UPADSTK.BIN"                      <HO>
610 Z=USR($9800,ADR(PM$),NR)                   <JK>
620 POKE $0601,TEIL                            <YG>
700 BLOAD "D:UKIPDRK.BIN":DKAD=$9500           <BT>
800 GOTO 7100                                  <MG>
999 --                                         <LD>
1500 PROC BUT:REPEAT :IF JTF:A=STICK(%1)
-14:ELSE :A=STRIG(%0):ENDIF :UNTIL A=%1:
PAUSE 4:ENDPROC                                <NB>
1699 --                                        <YB>
1700 PROC CRSGR8                               <PG>
1720 IF MODE=8 THEN POKE 87,8                  <XD>
1735 TRAP 1800                                 <QF>
1740 LOCATE X-1,Y,CR1:LOCATE X,Y-1,CR2:L
OCATE X,Y+1,CR3:LOCATE X+1,Y,CR4               <OV>
1760 COLOR 1-CR1:PLOT X-1,Y:COLOR 1-CR2:
PLOT X,Y-1:COLOR 1-CR3:PLOT X,Y+1:COLOR
1-CR4:PLOT X+1,Y:PAUSE 3                       <YT>
1780 COLOR CR1:PLOT X-1,Y:COLOR CR2:PLOT
 X,Y-1:COLOR CR3:PLOT X,Y+1:COLOR CR4:PL
OT X+1,Y                                       <PJ>
1800 COLOR COL:PAUSE 1                         <KG>
1820 IF MODE=8 THEN POKE 87,15:ENDIF           <JH>
1899 ENDPROC                                   <RD>
1999 REM _ FUNKTIONEN                          <MK>
2000 IF A=33 THEN 7000                         <QF>
2040 IF A=56 THEN GOSUB 12000                  <MD>
2060 FKTFG=%0:IF A=28 OR A=42 OR A=45 OR
 A=63 THEN 2090                                <WT>
2080 POKE N764,N255                            <DN>
2090 GOTO 2702                                 <SJ>
2500 REM ___ HAUPT                             <JM>
2600 FF=%0:CPF=%0:COL=ACOL:COLOR COL:POK
E N764,N255:GOSUB 12000                        <UO>
2620 IF GRS=10 THEN POKE 623,161               <AS>
2630 POKE $060F,%1                             <CI>
2700 # HAUPT                                   <JX>
2701 FKTFG=%1:A=PEEK(764):IF A<>N255 THE
N 2000                                         <IY>
2702 IF STICK(%1)=7 THEN 7000                  <EZ>
2703 A=PEEK(53279):IF A=%3 THEN JTF=JTF=
%0:POKE $0615,JTF:DPOKE $CB,X:POKE $CD,Y
:GOSUB 9800:FKTFG=%0                           <HZ>
2704 IF A=5 THEN JTMP=JTMP=%0:POKE $060D
,JTMP:GOSUB 9800:FKTFG=%0                      <LP>
2705 IF JTMP THEN POKE $060D,JTMP              <TG>
2710 IF JTF:P1=(DPEEK($0606) DIV TEIL-14
```

```
) DIV (APIX/%2):Y=(213-(DPEEK($0608) DIV
 TEIL)) DIV ASLI                              <QN>
2715 STR=STICK(%1)-14:X=P1:IF P1>212 THE
N X=XA:Y=YA                                   <MC>
2720 X=X*(X>%0):Y=Y*(Y>%0)                    <QH>
2721 IF X>XMAX THEN X=XMAX                    <KW>
2722 IF Y>YMAX THEN Y=YMAX                    <NM>
2725 DPOKE $CB,X:POKE $CD,Y                   <WB>
2730 ELSE :X=DPEEK($CB):Y=PEEK($CD):STR=
STRIG(%0):ENDIF                               <NG>
2750 IF MODE=9 THEN I=8/APIX:X=X DIV I*I <HC>
2770 IF GRS=8 AND STR=1 THEN EXEC CRSGR8 <ZO>
2800 IF FKTFG AND MODE<>%1 AND STR AND X
=XA AND Y=YA:GO# HAUPT:ENDIF :XA=X:YA=Y  <JZ>
2950 ON MODE GO# DRAW,LINE,LINEX,KREIS,K
REIS,RECHT,RECHT,FILL,COPY,TEXTG,ZOOM    <HN>
2995 REM ▲▲ ZEICHNEN                     <JK>
3000 # DRAW:IF STR THEN FF=%0:GO# HAUPT  <TT>
3020 IF FF:DRAWTO X,Y                    <BI>
3040 ELSE :PLOT X,Y:FF=%1                <CD>
3090 ENDIF :GO# HAUPT                    <FQ>
3190 REM ▲▲ PUNKTE                       <BT>
3200 # PUNKT:IF STR=%0 THEN PLOT X,Y     <HM>
3290 GO# HAUPT                           <FC>
3495 REM ▲▲ LINIEN                       <PM>
3500 # LINE:IF FF AND STR THEN MOVE SC2,
SC1,SCL:PLOT X1,Y1:DRAWTO X,Y:GO# HAUPT  <DU>
3600 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X1=X:Y1=Y:FF=%1:EXEC BUT:GO# HAUP
T                                        <IF>
3650 IF STR=%0 THEN FF=FF=%0:PLOT X1,Y1:
DRAWTO X,Y:EXEC BUT                      <NZ>
3690 GO# HAUPT                           <FK>
3699 REM ▲▲ LINIEN VERBUNDEN             <QK>
3700 # LINEX:IF PEEK(N764)=42 THEN FF=%0
:POKE N764,N255:MOVE SC2,SC1,SCL         <AH>
3710 IF FF AND STR THEN MOVE SC2,SC1,SCL
:PLOT X1,Y1:DRAWTO X,Y:GO# HAUPT         <MW>
3750 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:PLOT X,Y:X1=X:Y1=Y:FF=%1:EXEC BUT
:GO# HAUPT                               <RJ>
3800 IF STR=%0 THEN PLOT X1,Y1:DRAWTO X,
Y:X1=X:Y1=Y:MOVE SC1,SC2,SCL:EXEC BUT    <CX>
3990 GO# HAUPT                           <FQ>
3995 REM ▲▲ KREIS                        <WE>
4000 # KREIS:IF MODE=5:RD1=ABS(X-X1):RD2
=ABS(Y-Y1)                               <VR>
4020 ELSE :RD1=SQR((X-X1)^2+((Y-Y1)/(API
X/ASLI))^2):RD2=RD1*(APIX/ASLI):ENDIF    <LG>
4100 IF FF AND STR THEN MOVE SC2,SC1,SCL
:CIRCLE X1,Y1,RD1,RD2:GO# HAUPT          <OQ>
4200 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X1=X:Y1=Y:FF=%1:EXEC BUT:GO# HAUP
T                                        <HY>
4300 IF STR=%0 THEN FF=FF=%0:CIRCLE X1,Y
1,RD1,RD2:EXEC BUT                       <TM>
4490 GO# HAUPT                           <FH>
4495 REM ▲▲ RECHTECK                     <PQ>
4499 # RECHT                             <GI>
4500 IF FF AND STR THEN MOVE SC2,SC1,SCL
:PLOT X1,Y1:DRAWTO X1,Y:DRAWTO X,Y:DRAWT
O X,Y1:DRAWTO X1,Y1:GO# HAUPT            <NI>
4600 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X1=X:Y1=Y:FF=%1:EXEC BUT:GO# HAUP
T                                        <IG>
4700 IF STR=%0:FF=FF=%0:PLOT X1,Y1:DRAWT
O X1,Y:DRAWTO X1,Y1:DRAWTO X,Y1:DRAWTO X
1,Y1                                     <DS>
4720 IF MODE=7 THEN N=1-(2*(Y1>Y)):FOR I
=Y1 TO Y STEP N:PLOT X1,I:DRAWTO X,I:NEX
T I                                      <MA>
4750 EXEC BUT:ENDIF                      <BF>
4900 GO# HAUPT                           <EQ>
4995 REM ▲▲ AUSFUELLEN                   <XW>
5000 # FILL:IF PEEK(N764)=28 THEN MOVE S
C2,SC1,SCL:POKE N764,N255                <XN>
5080 IF STR=%0:IF GRS=8 THEN X=X DIV %2  <AS>
5100 PAINT X,Y:ENDIF                     <BT>
5200 GO# HAUPT                           <ED>
5500 REM ▲▲ KOPIEREN                     <AX>
5520 # COPY                              <CN>
5550 IF CPF=%0                           <TK>
5600 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X2=X:Y2=Y:FF=%1:GO# HAUPT         <IE>
5619 IF FF AND STR:MOVE SC2,SC1,SCL      <QM>
5620 PLOT X2,Y2:DRAWTO X2,Y:DRAWTO X,Y:D
RAWTO X,Y2:DRAWTO X2,Y2:GO# HAUPT:ENDIF  <EX>
5700 IF STR=%0:FF=%0:CPF=%1:X4=X:Y4=Y:X3
=X2:Y3=Y2:XD=ABS(X4-X3):YD=ABS(Y4-Y3):EX
EC BUT                                   <XA>
5710 IF XD<%1 OR YD<%1 THEN CPF=%0:FF=%1
:ENDIF                                   <CD>
5730 I=8/APIX:Z1=SC1:XDGER=INT(-XD/I)*-%
1:YDG=YD:MOVE SC2,SC1,SCL                <SF>
5750 ELSE                                <UB>
5760 IF X>XMAX-XD THEN X=XMAX-XD         <EQ>
5770 IF Y>YMAX-YD THEN Y=YMAX-YD         <JG>
5800 IF PEEK(N764)=63                    <LR>
5830 PLOT X3,Y3:DRAWTO X3,Y4:DRAWTO X4,Y
4:DRAWTO X4,Y3:DRAWTO X3,Y3              <YL>
5880 PLOT X,Y:DRAWTO X+XD,Y:DRAWTO X+XD,
Y+YD:DRAWTO X,Y+YD:DRAWTO X,Y            <SG>
5900 ELSE                                <TQ>
5910 Z=USR(DKAD+320,Z1-SC1+SC2,Z1,XDGER,
YDG,ABYT)                                <TV>
5920 I=8/APIX:XDGER=INT(-XD/I)*-%1       <JW>
5930 Q1=SC2+Y3*ABYT+X3 DIV I:Z1=SC1+Y*AB
YT+X DIV I:YDG=YD                        <NF>
5940 Z=USR(DKAD+320,Q1,Z1,XDGER,YDG,ABYT
)                                        <LB>
5980 IF STR=%0 THEN Z=USR(DKAD+320,Z1,Z1
-SC1+SC2,XDGER,YDG,ABYT):X3=X:Y3=Y:X4=X+
XD:Y4=Y+YD:ENDIF                         <AK>
5982 IF PEEK(N764)=42 THEN CPF=%0:MOVE S
C2,SC1,SCL:POKE N764,N255:ENDIF          <BO>
5990 GO# HAUPT                           <FS>
5999 REM ▲▲ TEXT                         <JV>
6000 # TEXTG:IF STR THEN FF=%1:MOVE SC2,
SC1,SCL:TEXT X,Y,TXT$:IF PEEK(N764)=45:G
OTO 6400:ENDIF :GO# HAUPT                <IO>
6300 IF STR=%0 THEN MOVE SC1,SC2,SCL:TEX
T X,Y,TXT$                               <IO>
6390 GO# HAUPT                           <FH>
6400 POKE N764,N255:TXT$="":N=%1         <VT>
6405 MOVE SC2,SC1,SCL                    <VO>
6410 DO :GET I:IF I=155:EXIT :ELSE :TXT$
(N,N)=CHR$(I):N=N+%1:MOVE SC2,SC1,SCL:TE
XT X,Y,TXT$:LOOP :ENDIF                  <ZK>
6480 POKE N764,N255                      <DZ>
6490 GO# HAUPT                           <FJ>
6499 REM ▲▲▲ ZOOM                        <RY>
6500 # ZOOM:ZMFG=%1                      <QO>
6520 CPF=1:XD=40/(APIX/ASLI)-%1:YD=23    <EG>
6550 IF FF=%0                            <HK>
6560 IF X>XMAX-XD THEN X=XMAX-XD         <EN>
6570 IF Y>YMAX-YD THEN Y=YMAX-YD         <JD>
6580 IF STR                              <LL>
6600 MOVE SC2,SC1,SCL                    <UY>
6630 PLOT X,Y:DRAWTO X+XD,Y:DRAWTO X+XD,
Y+YD:DRAWTO X,Y+YD:DRAWTO X,Y            <RO>
6680 ELSE :FF=%1:MOVE SC2,SC1,SCL        <PM>
6690 X1=X:Y1=Y:REM OGZ=GZ:GZ=1:GOSUB 800
0:GZ=OGZ                                 <KS>
6700 Z=USR(DKAD+400,%0,PUFAD,X,XD+%1,Y,Y
D+%1,APIX)                               <MQ>
6770 ENDIF :GO# HAUPT                    <GB>
6780 ENDIF                               <TX>
6800 I=X DIV 8:J=Y DIV 8:X=I*8:Y=J*8     <FW>
6820 COLOR PEEK(PUFAD+Y2*(XD+%1)+X2):TEX
T X2*8,Y2*8,"▲":COLOR COL:TEXT X,Y,"(CTL
 S)":X2=I:Y2=J                           <OF>
6850 IF STR=%0 THEN TEXT X,Y,"▲":POKE PU
FAD+J*(XD+%1)+I,COL                      <GH>
6900 IF PEEK(N764)=42:POKE N764,N255     <VM>
6910 MOVE SC2,SC1,SCL                    <VH>
```

Listing 1. »Grafik-Draw«



```
6920 Z=USR(DKAD+400,%1,PUFAD,X1,XD+%1,Y1
,YD+%1,APIX)                                 <MX>
6940 MOVE SC1,SC2,SCL:FF=%0:X=X1:Y=Y1:DP
OKE $CB,X:POKE $CD,Y:GOSUB 8000:GO# ZOOM <UO>
6995 ENDIF :GO# HAUPT                        <HF>
7000 REM ▲▲▲ MENU                            <HC>
7050 IF FF=%0 AND MODE<>11 AND MODE<>9 T
HEN MOVE SC1,SC2,SCL                         <YQ>
7100 POKE $060D,%0:POKE $060F,%0:POKE $0
618,%0                                       <NS>
7200 GRAPHICS %0:POKE 559,%0:POKE 53249,
%0:POKE 53250,%0:POKE 710,10:POKE 709,0:
POKE 712,80                                  <XG>
7210 POKE N764,N255                          <DB>
7250 MPS=MODE:POKE $060C,191:POKE $CD,MO
DE*6:Y=MODE*8+56:GOSUB 9000                  <FY>
7300 POKE 752,%1:? "{ESC CTL <}{CTL Q}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 E}"                                        <BB>
7350 ? "{SHIFT =}▲▲▲▲▲▲▲▲▲▲▲▲Grafik▲Draw
":POSITION 37,%1:? "{SHIFT =}"               <PU>
7380 ? "{SHIFT =}▲▲▲▲(c)▲1987▲▲Klaus▲Kra
ssowski▲":POSITION 37,%2:? "{SHIFT =}"       <BJ>
7450 ? "{CTL A}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL W}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL C}"                        <YQ>
7600 ? "{SHIFT =}▲Zeichnen▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <ZN>
7605 ? "{SHIFT =}▲Linie▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <JC>
7610 ? "{SHIFT =}▲Linie▲folgend▲▲▲{SHIFT
 =}"                                         <KY>
7615 ? "{SHIFT =}▲Kreis▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <QN>
7620 ? "{SHIFT =}▲Ellipse▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <EN>
7625 ? "{SHIFT =}▲Rechteck▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <UV>
7630 ? "{SHIFT =}▲Rechteck▲gef.▲▲▲{SHIFT
 =}"                                         <UD>
7640 ? "{SHIFT =}▲Ausfuellen▲▲▲▲▲▲{SHIFT
 =}"                                         <OZ>
7650 ? "{SHIFT =}▲Kopieren▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <FV>
7652 ? "{SHIFT =}▲Text▲▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <HW>
7654 ? "{SHIFT =}▲Zoom▲▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                         <HB>
7656 ? "{CTL A}▲Farben▲▲▲▲▲▲▲▲▲{CTL R}{C
TL S}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}
{ESC CTL =}{ESC CTL +}{SHIFT =}{ESC CTL
-}"                                          <FO>
7660 ? "{SHIFT =}▲Grafik▲Stufe▲▲▲▲:[";GR
S;"]"                                        <EG>
7670 ? "{SHIFT =}▲Loeschen▲▲▲▲▲▲▲▲{CTL A
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL X}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL E}{ESC CT
L =}{ESC CTL +}{SHIFT =}{ESC CTL -}"         <XP>
7680 ? "{SHIFT =}▲Disk▲▲▲▲▲▲▲▲▲▲▲▲:";GRN
$                                            <CH>
7690 ? "{SHIFT =}▲Drucken▲▲▲▲▲▲▲▲▲{CTL A
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL D}{ESC CT
L =}{ESC CTL +}{SHIFT =}{ESC CTL -}"         <SV>
7692 ? "{SHIFT =}▲Zeichengeraet▲▲▲:";        <GI>
7695 IF JTF:? "Touch▲Tablet":ELSE :? "Jo
ystick":ENDIF                                <KU>
7698 ? "{CTL Z}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL X}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL C}":? :POKE 752,%0                <ST>
7700 DO :IF JTF=%0:Y=PEEK($CD):A=Y DIV 6 <WO>
7705 ELSE :Y=200-PADDLE(%3):Y=Y*(Y>%0):A
=Y DIV 10:PAUSE %1:ENDIF                     <QK>
7710 ST1=STRIG(%0):ST2=STICK(%1)-14          <XQ>
7714 IF A AND MPS<>A:MPS=A                   <FB>
7716 IF MPS>17:MPS=17:POKE $CD,MPS*6:ELS
E                                            <NW>
7718 SOUND %0,MPS*8+20,10,LAUTS:PAUSE %1 <JW>
7720 Y=MPS*8+56:GOSUB 9000                   <WW>
7725 ENDIF :SOUND %0,%0,%0,%0:ENDIF :IF
ST1=%0 OR ST2=%0 THEN EXIT                   <GC>
7730 LOOP                                    <YX>
7770 IF MPS<12 THEN MODE=MPS                 <LB>
7800 IF MPS=13 THEN GOSUB 14000              <QX>
7810 IF GRS=5 AND (MODE=9 OR MODE=11 OR
MPS=16) THEN GOSUB 9800:MODE=1:GOTO 7100 <OV>
7820 IF MPS=14 THEN ? "Bild▲Loeschen▲[J/
N]▲";:GET I:IF I<>ASC("J") THEN 7300         <MC>
7830 IF MPS=10 THEN ? "Text▲->▲:";:INPUT
 TXT$                                        <YF>
7850 IF MPS=15 THEN GOSUB 17000              <VU>
7870 IF MPS=17 THEN JTF=JTF=%0:POKE $061
5,JTF:GOTO 7300                              <EV>
7880 IF MODE=9 THEN POKE $0618,%1            <RE>
7900 IF MPS=16 THEN GOSUB 21000              <QK>
7940 REM                                     <UI>
7950 POKE 53249,%0:POKE 53250,%0             <PT>
7960 POKE 559,%0:GRAPHICS GRS+16:SC1=DPE
EK(88)                                       <EA>
7962 GOSUB 8000:GOSUB 15000                  <JA>
7965 POKE $060C,YMAX:MOVE SC2,SC1,SCL        <FA>
7970 IF GRS=8 THEN POKE 712,%2               <RJ>
7975 IF MPS=14 THEN CLS #6:MOVE SC1,SC2,
SCL                                          <JC>
7980 IF MPS=16 AND DRFG THEN GOSUB 20000
:GOTO 7000                                   <OR>
7985 IF MPS=12 THEN GOSUB 12500              <VO>
7990 IF MPS=15 THEN ON LSF GOSUB 18000,1
8500:GOSUB 15000:MOVE SC1,SC2,SCL:GOTO 7
000                                          <OJ>
7995 GOTO 2600                               <SM>
8000 REM ▲▲▲ PM STEUERUNG                    <EH>
8050 NR=%0:RESTORE 8100+GZ*10                <ER>
8060 DO :READ A:IF A=-1 THEN EXIT            <AL>
8080 NR=NR+%1:PM$(NR,NR)=CHR$(A)             <SA>
8090 LOOP :POKE $060E,NR                     <YT>
8110 DATA▲0,0,48,48,72,72,72,72,48,48,0,
-1                                           <FI>
8120 DATA▲0,0,32,32,80,80,32,32,0,-1         <WB>
8130 DATA▲0,0,48,48,204,48,48,0,-1           <OQ>
8140 DATA▲0,0,24,24,102,24,24,0,-1           <RS>
8150 DATA▲0,0,48,48,204,48,48,0,-1           <OW>
8160 DATA▲0,0,32,32,80,32,32,0,0,0,-1        <ZD>
8170 DATA▲0,0,0,0,0,-1                       <DI>
8200 POKE 559,62:POKE 705,15                 <ON>
8210 I=PEEK(106)-40:PMAD=I*256+1280          <XA>
8220 POKE 54279,I:POKE 53277,%3              <QP>
8260 POKE 53257,%0:REM PLAYER 1 !!           <LB>
8262 POKE 53258,%1                           <EC>
8300 POKE PMAD,%0:MOVE PMAD,PMAD+%1,N255 <CR>
8310 POKE PMAD+N255,%0:MOVE PMAD+N255,PM
AD+256,N255                                  <QG>
8320 POKE PMAD+N255+20,N255:MOVE PMAD+N2
55+20,PMAD+N255+21,10                        <LT>
8350 POKE 53250,48                           <NU>
8490 RETURN                                  <QJ>
8990 REM ▲▲ PM MENU                          <FM>
9000 POKE PMAD,%0:MOVE PMAD,PMAD+%1,512      <JK>
9050 POKE 559,62:POKE 53277,%3:POKE 623,
%1                                           <PN>
9055 IF MPS<12:POKE 705,186:POKE 706,186
:ELSE :POKE 705,42:POKE 706,42:ENDIF         <NW>
9070 POKE 53257,%3:POKE 53258,%3:POKE 53
249,62:POKE 53250,90                         <WR>
9100 MOVE ADR(PM2$),PMAD+Y,8                 <NN>
```

```
9120 MOVE ADR(PM2$),PMAD+256+Y,8          <TZ>
9190 RETURN                               <QE>
9499 REM ▲ PM AUS                         <VU>
9500 POKE 710,16:POKE 709,12:POKE 705,16
:POKE 706,16:RETURN                       <HP>
9799 REM ▲ TON                            <JO>
9800 SOUND %0,90,10,LAUTS:PAUSE 8:SOUND
%0,%0,%0,%0:RETURN                        <LR>
10000 REM ▲▲▲ INIT                        <GE>
10020 DIM PM$(30),PM2$(13),GRSII$(7700),
PUF$(1000),CPF$(320)                      <QJ>
10030 DIM A$(15),GRN$(15),TXT$(40)        <TO>
10040 DIM GCOL(7,8)                       <VH>
10050 SC2=ADR(GRSII$):PMSTR=ADR(PM$)      <YL>
10080 DIM GR$(6),KLR$(40)                 <VW>
10150 PM2$="{ESC CTL >}":PM2$(12)="{ESC
CTL >}":PM2$(%2)=PM2$                     <ON>
10190 GRSII$="{CTL ,}":GRSII$(1300)="{CT
L ,}":GRSII$(%2)=GRSII$:PUF$=GRSII$       <PE>
10200 KLR$="▲":KLR$(40)="▲":KLR$(%2)=KLR
$:GRN$=KLR$                               <MQ>
10220 PUFAD=ADR(PUF$)                     <HR>
10230 GR$=CHR$(27):GR$(%2)="K"            <VK>
10330 N255=255:N764=764                   <SV>
10350 COL=%1:ACOL=COL:DFMT=%1:DRFG=%0:JT
F=%0:JTMP=%0:INVFG=%0:LAUTS=4:TEIL=%2     <XH>
10400 MODE=%1                             <QC>
10410 GZ=6:GOSUB 14100:GOSUB 14500        <RV>
10490 RETURN                              <UT>
12000 REM ▲▲ FARB WAHL                    <AC>
12020 GOSUB 9800:A=CM:IF GRS=9 OR GRS=11
 THEN A=16                                <ZD>
12030 IF GRS=8 AND MODE=8 THEN POKE 87,1
5:A=4:X=X DIV %2                          <MV>
12080 ACOL=COL                            <IJ>
12100 COL=COL+%1-(COL+%1)*(COL>=A-%1)     <CV>
12120 COLOR COL                           <SC>
12200 IF GRS>8 AND GRS<12 OR (GRS=8 AND
MODE=8):POKE 53250,%0:MOVE SC1,ADR(CPF$)
,320:TEXT %0,%0,"▲▲"                      <XE>
12220 PAUSE 30:MOVE ADR(CPF$),SC1,320:EL
SE :POKE 706,GCOL(CMEN*%2+%1,COL)         <XV>
12300 ENDIF :POKE N764,N255               <BG>
12390 RETURN                              <UU>
12490 REM ▲▲ FARB MENU                    <KE>
12500 A=CM:A$="▲▲▲▲▲▲▲▲":IF GRS=9 OR GRS
=11 THEN A=16                             <YG>
12600 FOR I=%0 TO A-%1                    <AQ>
12640 COLOR I:TEXT %0,I*12,A$(%1,8/APIX)  <YD>
12660 NEXT I:COL=%0:TEXT 64/APIX,%0,"{ES
C CTL +}"                                 <FX>
12700 DO :A=PEEK(53279):J=CMEN*%2         <OP>
12720 IF A=6 THEN TEXT 64/APIX,COL*12,"▲
":COL=(COL+%1)*(COL<CM-%1):TEXT 64/APIX,
COL*12,"{ESC CTL +}":GOSUB 9800           <LH>
12740 I=GCOL(J+%1,COL):H=I MOD 16:F=I-H   <HB>
12760 IF A=%3 THEN F=(F+16)*(F<240):GCOL
(J+%1,COL)=F+H:POKE GCOL(J,COL),F+H:PAUS
E 10                                      <JC>
12800 IF A=5 THEN H=(H+%2)*(H<14):GCOL(J
+%1,COL)=F+H:POKE GCOL(J,COL),F+H:PAUSE
10                                        <AC>
12880 IF PEEK(N764)=28 THEN POKE N764,N2
55:EXIT                                   <LO>
12900 LOOP                                <WT>
12990 MOVE SC2,SC1,SCL:RETURN             <BU>
14000 REM ▲▲ GRAPHIC WAHL                 <UI>
14020 ACOL=%0:GOSUB 9500:? "{ESC CTL <}{
ESC CTL =}"                               <RU>
14050 ? "Welche▲Grafik-Stufe?":? :? "1▲=
▲[▲5]":? "2▲=▲[▲7]":? "3▲=▲[▲9]":? "4▲=▲
[10]":? "5▲=▲[11]":? "6▲=▲[15]"           <JL>
14060 ? "7▲=▲[▲8]"                        <ZH>
14070 REPEAT :GET I:I=I-48:UNTIL I>%0 AN
D I<8                                     <IL>
14080 GZ=I                                <CR>
14100 RESTORE 14390+GZ*10:READ GRS,SCL,X
MAX,YMAX,ABYT,APIX,ASLI,CMEN,CM           <IS>
14210 DPOKE $060A,XMAX:POKE $060C,YMAX    <EG>
14220 POKE $0611,APIX:POKE $0612,ASLI     <AV>
14290 RETURN                              <UV>
14392 REM GRS,SCL,XMAX,YMAX,ABYT,APIX,AS
LI                                        <NV>
14400 DATA▲5,960,79,47,20,4,4,1,4         <IN>
14410 DATA▲7,3840,159,95,40,2,2,1,4       <HT>
14420 DATA▲9,7680,79,191,40,4,1,3,1       <JI>
14430 DATA▲10,7680,79,191,40,4,1,2,9      <FD>
14440 DATA▲11,7680,79,191,40,4,1,3,1      <WA>
14450 DATA▲15,7680,159,191,40,2,1,1,4     <NL>
14460 DATA▲8,7680,319,191,40,1,1,0,2      <QD>
14490 REM ▲▲ COLORS                       <CP>
14500 RESTORE 14600                       <LG>
14510 FOR N=%0 TO %3:READ J               <PG>
14520 FOR I=%0 TO J-%1                    <HO>
14530 READ A,B:GCOL(N*%2,I)=700+A         <FW>
14540 GCOL(N*%2+%1,I)=B                   <ZQ>
14550 NEXT I:NEXT N                       <UB>
14590 RETURN                              <VE>
14600 DATA▲2,10,0,9,15                    <WB>
14610 DATA▲4,12,0,8,186,9,150,10,40       <DN>
14620 DATA▲9,4,8,5,14,6,184,7,150,8,32,9
,176,10,90,11,80,12,60                    <EQ>
14630 DATA▲1,12,8                         <UB>
15000 REM ▲▲ FARBEN SETZEN                <MR>
15010 FOR I=%0 TO CM-%1:J=CMEN*%2         <CH>
15050 POKE GCOL(J,I),GCOL(J+%1,I)         <XJ>
15100 NEXT I:RETURN                       <AM>
17000 REM ▲▲ FILENAME                     <NQ>
17005 TRAP 7200:GOSUB 9500:? "{ESC CTL <
}{ESC CTL =}Disk▲Menu"                    <HW>
17010 DO :LSF=%0:?                        <PY>
17020 ? "[D]▲Directory"                   <MA>
17030 ? "[L]▲Laden"                       <IK>
17040 ? "[S]▲Speichern"                   <ZQ>
17050 ? "[{ESC ESC}{ESC ESC}]▲ENDE":?     <FC>
17060 GET I                               <TF>
17100 IF I=76 THEN LSF=%1:EXIT            <RS>
17110 IF I=83 THEN LSF=%2:EXIT            <RU>
17120 IF I=68 THEN ? :? :DIR              <XX>
17125 IF I=27 THEN EXIT                   <OB>
17130 LOOP :IF I=27 THEN 17300            <EW>
17140 PUT I+128:? :? :IF LSF=%2 THEN ? ,
"{ESC CTL *}";GRN$;"{ESC CTL -}"          <NR>
17150 INPUT "Filename▲{ESC ESC}{ESC CTL
*}";A$                                    <OR>
17160 A=INSTR(A$,":"):IF A=%2 OR A=%3 TH
EN GRN$=A$:RETURN                         <DI>
17180 GRN$(%1)="D:":GRN$(%3)=A$           <MM>
17300 RETURN                              <TU>
18000 REM ▲▲▲ LADEN                       <RW>
18100 TRAP 18490:OPEN #4,4,0,GRN$         <EX>
18150 ACOL=%0:CLS #6:BGET #4,SC1,SCL      <DO>
18200 GET #4,I:IF I=75                    <HR>
18250 GET #4,GZ:GOSUB 14100               <QZ>
18300 GET #4,CM                           <ZD>
18320 FOR I=%0 TO CM-%1:J=CMEN*%2         <DA>
18340 GET #4,A:GET #4,B                   <KP>
18350 GCOL(J,I)=A+700:GCOL(J+%1,I)=B      <IV>
18400 NEXT I                              <GL>
18490 ENDIF :CLOSE #4:RETURN              <RZ>
18500 REM ▲▲▲ SPEICHERN                   <QP>
18600 TRAP 18900:OPEN #4,8,0,GRN$         <GZ>
18700 BPUT #4,SC1,SCL                     <ZG>
18800 PUT #4,75:PUT #4,GZ:PUT #4,CM       <CY>
18820 FOR I=%0 TO CM-%1:J=CMEN*%2         <DP>
18830 A=GCOL(J,I)-700:B=GCOL(J+%1,I)      <ET>
18850 PUT #4,A:PUT #4,B                   <IR>
18860 NEXT I                              <HV>
18900 CLOSE #4:RETURN                     <FK>
20000 REM ▲▲▲ DRUCK ROUTINE               <OY>
20100 TRAP 20600:OPEN #3,8,0,"P:"         <FN>
20120 IF INVFG THEN Z=USR(DKAD+280,YMAX+
%1)                                       <GY>
20150 ? #3;"{ESC ESC}A{CTL H}"            <BH>
```

Listing 1. »Grafik-Draw« (Fortsetzung)

```
20200 FOR N=SC1 TO SC1+SCL-320 STEP 320  <FB>
20250 IF PEEK(N764)=126 THEN POKE N764,N
255:EXIT                                 <KP>
20300 Z=USR(DKAD,DFMT,N,SC2)             <CO>
20350 IF DFMT=%1 THEN ? #3;KLR$(%1,DPS);
GR$;GRSII$(%1,320)                       <RU>
20400 IF DFMT=%2 THEN ? #3;GR$;GRSII$(%1
,640):? #3;GR$;GRSII$(641,1280)          <CN>
20450 NEXT N                             <IY>
20500 IF INVFG THEN Z=USR(DKAD+280,YMAX+
%1)                                      <HC>
20550 MOVE SC1,SC2,SCL                   <UX>
20600 CLOSE #3:RETURN                    <DY>
21000 REM ▲▲▲ DRUCK FORMAT               <AW>
21050 GOSUB 9500                         <LV>
21100 REPEAT :? "{ESC CTL <}{ESC CTL =}{
ESC CTL =}DRUCK▲Format":? "{ESC CTL =}[1
]▲Halbe▲DIN▲A4▲Breite":? "[2]▲Ganze▲DIN▲
A4▲Breite":?                             <GH>
21120 GET A:PUT A+128:A=A-48             <SB>
21150 UNTIL A>%0 AND A<%3                <UF>
21200 DFMT=A                             <IC>
21250 IF DFMT=%1 THEN GR$(3,3)=CHR$(64):
GR$(4,4)=CHR$(1)                         <MO>
21450 IF DFMT=%2 THEN GR$(3,3)=CHR$(128)
:GR$(4,4)=CHR$(2):GOTO 23000             <UC>
21750 REPEAT :? "{ESC CTL <}{ESC CTL =}{
ESC CTL =}DRUCK▲Position":? "{ESC CTL =}
[1]▲Links":? "[2]▲Mitte":? "[3]▲Rechts":
? "[4]▲Individuel"                       <LV>
21800 ? :GET A:PUT A+128:A=A-48:UNTIL A>
%0 AND A<5                               <GN>
21850 IF A<>4:DPS=(A-%1)*19+%1           <VE>
22150 ELSE :? :REPEAT :? "{ESC CTL -}{ES
C SHIFT DEL}Position▲[1-40]▲:";:INPUT DP
S:UNTIL DPS>%0 AND DPS<41                <TP>
23000 ENDIF                              <RZ>
23020 ? :? "{ESC CTL =}Invers▲Drucken▲▲[
J/N]▲";:GET I:IF I=ASC("J"):INVFG=%1:ELS
E :INVFG=%0:ENDIF                        <ZP>
23500 ? :? "{ESC CTL =}Starte▲Drucken▲▲[
J/N]▲";:GET I                            <DL>
23510 IF I=ASC("J"):DRFG=%1:ELSE :DRFG=%
0:INVFG=%0:ENDIF                         <WE>
24000 RETURN                             <TG>
```

Listing 1. »Grafik-Draw« (Schluß)

```
Programmname :UPADSTK.BIN
Länge        :400 Bytes
-----------------------------------

0000:FF FF 00 98 86 99 68 68 <0A>
0008:85 CF 68 85 CE 68 68 8D <5D>
0010:0E 06 A9 00 8D 0F 06 A9 <03>
0018:14 8D 01 06 A2 98 A0 20 <90>
0020:A9 07 20 5C E4 60 AD 15 <83>
0028:06 F0 55 AD 72 02 18 6D <DD>
0030:02 06 8D 02 06 90 03 EE <CA>
0038:03 06 AD 73 02 18 6D 04 <57>
0040:06 8D 04 06 90 03 EE 05 <F2>
0048:06 EE 00 06 AD 00 06 CD <65>
0050:01 06 D0 2C AD 02 06 8D <15>
0058:06 06 AD 03 06 8D 07 06 <19>
0060:AD 04 06 8D 08 06 AD 05 <F7>
0068:06 8D 09 06 A9 00 8D 00 <66>
0070:06 8D 02 06 8D 03 06 8D <40>
0078:04 06 8D 05 06 4C BB 98 <4C>
0080:AE 0D 06 E0 02 B0 6C AE <2F>
0088:78 02 8A 29 04 D0 15 A5 <57>
0090:CC D0 04 A5 CB F0 0D 38 <21>
0098:A5 CB E9 01 85 CB A5 CC <B9>
00A0:E9 00 85 CC 8A 29 08 D0 <3B>
00A8:06 E6 CB D0 02 E6 CC 8A <64>
00B0:29 01 D0 06 A5 CD F0 02 <3A>
00B8:C6 CD 8A 29 02 D0 02 E6 <CB>
00C0:CD A5 CC CD 0B 06 90 18 <3E>
00C8:F0 03 4C D0 98 A5 CB CD <C5>
00D0:0A 06 90 0C F0 0A AD 0A <15>
00D8:06 85 CB AD 0B 06 85 CC <A6>
00E0:AE 0C 06 E4 CD B0 02 86 <6C>
00E8:CD AE 0D 06 F0 05 A2 02 <D7>
00F0:8E 0D 06 AE 0F 06 D0 03 <C9>
00F8:4C 62 E4 18 AD 10 06 69 <4F>
0100:1C A8 A2 00 8A 99 00 9D <BA>
0108:C8 E8 EC 0E 06 D0 F6 A5 <C7>
0110:CD 8D 16 06 AE 12 06 E0 <E1>
0118:02 D0 07 18 2E 16 06 4C <83>
0120:27 99 E0 04 D0 07 18 2E <18>
0128:16 06 2E 16 06 18 AD 16 <DF>
0130:06 8D 10 06 69 1C AA A0 <6B>
0138:00 B1 CE 9D 00 9D E8 C8 <04>
0140:CC 0E 06 D0 F4 A5 CB 8D <C2>
0148:13 06 A5 CC 8D 14 06 AE <15>
0150:11 06 E0 02 F0 15 E0 01 <11>
0158:D0 0A 18 6E 14 06 6E 13 <55>
0160:06 4C 65 99 18 2E 13 06 <40>
0168:4C 65 99 AD 18 06 F0 10 <28>
0170:EE 13 06 18 4E 13 06 4E <79>
0178:13 06 0E 13 06 0E 13 06 <84>
0180:18 AD 13 06 69 2E 8D 01 <D4>
0188:D0 4C 62 E4 00 00 00 00 <23>
01E0:60 AD 84 06 9D 42 03 A9 <67>
01E8:00 9D 48 03 9D 49 03 A5 <00>
01F0:E2 F0 04 A0 00 B1 E0 20 <EC>
01F8:56 E4 A4 E2 D0 02 91 E0 <C0>
0200:E6 E0 D0 02 E6 E1 C6 D7 <A0>
0208:D0 D1 AD 82 06 85 D7 E6 <4C>
0210:D6 CE 83 06 D0 BD A5 E2 <40>
0218:F0 01 60 AD 86 06 85 E0 <50>
0220:AD 87 06 85 E1 AD 82 06 <6C>
0228:85 D7 A5 E3 8D 83 06 A5 <A5>
0230:E5 38 E9 01 0A AA BD 49 <9B>
0238:97 8D 6D 97 BD 4A 97 8D <CF>
0240:6E 97 A5 58 8D 80 06 A5 <76>
0248:59 8D 81 06 4C 51 97 6F <CE>
0250:97 8D 97 00 00 78 97 AD <71>
0258:80 06 85 D4 AD 81 06 85 <67>
0260:D5 18 A5 D4 69 A0 85 D8 <4B>
0268:A5 D5 69 00 85 D9 A0 00 <D1>
0270:B1 E0 4C 6F 97 F0 02 A9 <56>
0278:FF 85 D6 4C A0 97 85 E2 <E3>
0280:85 D6 06 E2 06 E2 06 E2 <85>
0288:06 E2 A5 E2 05 D6 85 D6 <3B>
0290:4C A0 97 85 E2 85 D6 A0 <5E>
0298:03 06 E2 06 E2 A5 E2 05 <A6>
02A0:D6 85 D6 88 D0 F3 A6 E5 <A0>
02A8:A0 00 91 D4 91 D8 A0 28 <99>
02B0:91 D4 91 D8 A0 50 91 D4 <3B>
02B8:91 D8 A0 78 91 D4 91 D8 <75>
02C0:E6 D4 D0 02 E6 D5 E6 D8 <ED>
02C8:D0 02 E6 D9 CA D0 D9 E6 <B4>
02D0:E0 D0 02 E6 E1 C6 D7 D0 <33>
02D8:95 AD 82 06 85 D7 18 AD <47>
02E0:80 06 69 40 8D 80 06 AD <72>
02E8:81 06 69 01 8D 81 06 CE <30>
02F0:83 06 F0 03 4C 51 97 60 <12>
02F8:00 00 00 00 00 00 00 00 <FC>
0300:00 00 00 00 00 00 00 00 <06>
```

Listing 2. Bitte mit AMPEL eingeben

```
Programmname :UKIPDRK.BIN
Länge        :776 Bytes
-----------------------------------

0000:FF FF 00 95 FF 97 68 68 <96>
0008:68 8D 83 06 68 85 E1 85 <9C>
0010:D8 68 85 E0 85 D7 68 85 <BA>
0018:E3 68 85 E2 A9 08 8D 82 <7A>
0020:06 AD 83 06 C9 02 D0 04 <11>
0028:18 6E 82 06 18 A5 E2 69 <B4>
0030:80 85 E4 A6 E3 90 01 E8 <AF>
0038:E8 E8 86 E5 A2 80 86 D4 <31>
0040:86 D5 A2 00 86 D9 8E 80 <D9>
0048:06 86 D6 8E 81 06 A0 00 <E5>
0050:B1 E0 25 D4 F0 1A 18 A5 <3F>
0058:D5 65 D6 85 D6 AD 83 06 <9D>
0060:C9 02 D0 16 18 66 D5 A5 <59>
0068:D5 65 D6 85 D6 18 90 0A <79>
0070:AD 83 06 C9 02 D0 03 18 <62>
0078:66 D5 66 D5 18 A5 E0 69 <F6>
0080:28 85 E0 90 02 E6 E1 E8 <4C>
0088:EC 82 06 D0 C1 A2 02 AC <49>
0090:81 06 C0 01 F0 12 A0 00 <E3>
0098:A5 D6 91 E2 E6 E2 D0 02 <8D>
00A0:E6 E3 18 90 0F 18 90 A6 <E6>
00A8:A0 00 A5 D6 91 E4 E6 E4 <B6>
00B0:D0 02 E6 E5 AD 83 06 C9 <6C>
00B8:02 D0 15 CA D0 D1 A9 80 <90>
00C0:85 D5 A2 00 86 D6 EE 81 <8D>
00C8:06 AC 81 06 C0 02 D0 D5 <E2>
00D0:18 66 D4 A5 D7 85 E0 A5 <7A>
00D8:D8 85 E1 A9 80 85 D5 A2 <3A>
00E0:00 86 D6 8E 81 06 E6 D9 <B8>
00E8:A4 D9 C0 08 D0 B7 85 D4 <A9>
00F0:86 D9 18 E6 D7 E6 E0 D0 <29>
00F8:04 E6 D8 E6 E1 EE 80 06 <93>
0100:AC 80 06 C0 28 D0 9E 60 <50>
0108:00 00 00 00 00 00 00 00 <0A>
0110:00 00 00 00 00 00 00 00 <12>
0118:00 00 00 00 00 00 68 68 <52>
0120:68 85 D6 A5 58 85 E0 A5 <06>
0128:59 85 E1 A0 00 A2 28 B1 <E6>
0130:E0 49 FF 91 E0 E6 E0 D0 <95>
0138:02 E6 E1 CA D0 F1 C6 D6 <21>
0140:D0 EB 60 00 00 00 68 68 <29>
0148:85 E1 68 85 E0 68 85 E3 <B8>
0150:68 85 E2 68 68 85 D4 68 <93>
0158:68 85 D5 68 68 85 D6 A6 <27>
0160:D5 A4 D4 88 B1 E0 91 E2 <0B>
0168:88 10 F9 18 A5 E0 65 D6 <1E>
0170:85 E0 90 02 E6 E1 18 A5 <66>
0178:E2 65 D6 85 E2 90 02 E6 <DE>
0180:E3 CA D0 DD 60 00 00 00 <B9>
0188:00 00 00 00 00 00 00 00 <8A>
0190:00 00 00 00 00 00 68 68 <CB>
0198:68 85 E2 68 85 E1 8D 87 <34>
01A0:06 68 85 E0 8D 86 06 68 <4C>
01A8:85 D5 68 85 D4 68 68 8D <92>
01B0:82 06 85 D7 68 68 85 D6 <35>
01B8:68 68 8D 83 06 85 E3 68 <12>
01C0:68 85 E5 A5 E2 D0 07 A9 <14>
01C8:07 8D 84 06 10 05 A9 0B <73>
01D0:8D 84 06 A5 D4 85 55 A5 <43>
01D8:D5 85 56 A5 D6 85 54 A2 <D7>
```

Listing 3. Bitte mit AMPEL eingeben

3D-Grafik in Turbo Basic

Dreidimensionale Computergrafiken sind immer häufiger Bestandteil von Film und Fernsehen. Mit dem Atari lassen sich bereits ansehnliche Grafiken berechnen und darstellen.

Nicht nur in der Industrie, in der Entwicklung oder im CAD-Bereich, sondern auch in Film und Fernsehen, besonders in der Werbung, bedient man sich immer mehr der »Bilder aus dem Computer«. Das ist kein Wunder, denn die Kosten für computeranimierte Filme sind drastisch gesunken. Computer, die solche Leistungen bringen, sind meistens teure Grafik-Workstations oder Supercomputer wie die »Cray« und besitzen neben schnellen Programmen spezielle Hardware. Dadurch wird die Grafik wesentlich schneller und viele Grafik-Routinen überflüssig. Eins steht auf jeden Fall fest: Die Berechnung von dreidimensionalen Grafiken mit verdeckten Kanten, Schattierung und Lichtreflexen braucht sehr viel Speicher und eine Menge Zeit.

Mit diesem Beitrag wollen wir nicht die mathematischen Grundlagen der dreidimensionalen Grafik erklären. Dafür behandeln wir das sogenannte »Clipping« oder »Windowing«, verdeckte Kanten und die Berechnung von Schlagschatten einfacher Objekte. Da bestimmte Unterprogramme immer wieder Verwendung finden, werden die einzelnen Prozeduren gesondert in einem Satz erläutert.

Steigen wir mit dem »Clipping« ein. Dahinter verbirgt sich nichts weiter als das Kappen von Linien, die über den Bildschirmrand hinausgehen. Ganz so einfach ist die Sache leider nicht. Ein wichtiger Faktor ist nämlich der Winkel, unter dem die Linie den Bildrand schneidet. Die Clipping-Routine berechnet nun im Prinzip die Länge und den Winkel des nicht sichtbaren Teils einer Linie gegenüber dem Bildrand und liefert als Ergebnis die erste wieder sichtbare Koordinate dieser Linie.

Der hier verwendete Clipping-Algorithmus wurde in Amerika von »Evans und Sutherland«, Pionieren der Computergrafik, entwickelt. Dieses Unternehmen liefert Computersimulationen hauptsächlich für militärische Zwecke.

Doch nun zum Programm: Innerhalb der Routine wird mit Hilfe der Prozedur »CODE« zunächst ein 4-Bit-Code berechnet, wobei die Bits folgende Bedeutung haben:
Bit 1: Punkt ist links vom Bildrand
Bit 2: Punkt ist rechts vom Bildrand
Bit 3: Punkt ist unterhalb vom Bildrand
Bit 4: Punkt ist oberhalb vom Bildrand

Die Variablen, die diese Bitkombinationen enthalten, sind C, C1 und C2. CZ dient hier nur als Tauschvariable. Die Bits sind entsprechend den Bitpositionen innerhalb eines Byte in den Variablen enthalten. Die Prozedur »CLIP« berechnet dann Länge und Winkel der verdeckten Linie(n). Dieser Prozeß wird solange durchlaufen, bis die Prozedur erkennt, daß die ganze Linie unsichtbar ist (Bedingung: VIS=0). Wenn Teile der Linie oder die ganze Linie sichtbar sind, so lautet die Bedingung VIS=1. Haben alle vier Bits den Wert 0, befindet sich der zu setzende Punkt oder die Linie innerhalb des Bildes und kann gesetzt werden. Die Variablen, die die Bildschirmgrenzen festsetzen, heißen:

CLIPYB	für den unteren Rand
CLIPYT	für den oberen Rand
CLIPXL	für den linken Rand
CLIPXR	für den rechten Rand

Durch Verändern dieser Variablen (im Listing Zeile 40 und 50) wird jeder beliebige Bildschirmbereich bestimmt. Das Objekt oder sichtbare Teile davon werden nur innerhalb dieses Fensters dargestellt. Obwohl wir hier mit dreidimensionalen Objekten arbeiten, ist die Clipping-Routine im Prinzip für zweidimensionale Objekte gedacht. Da auf dem Bildschirm aber zweidimensional abgebildet wird, kann die Routine verwendet werden. Es existiert zwar auch ein Cohen-Sutherland-Algorithmus, der dreidimensional arbeitet. Er ist aber nur von Interesse, wenn man sich in das Objekt »hinein«begeben will. Damit dieses dennoch mit der hier verwendeten Routine bedingt möglich ist, wird in der Prozedur »CALCPOINT« die Z-Koordinate eines Punktes automatisch auf 1 gesetzt, wenn diese vorher kleiner war. Damit wird verhindert, daß eventuell »vor« dem Bildschirm liegende Objekte durch falsche Berechnung ein Linienchaos auf dem Schirm erzeugen.

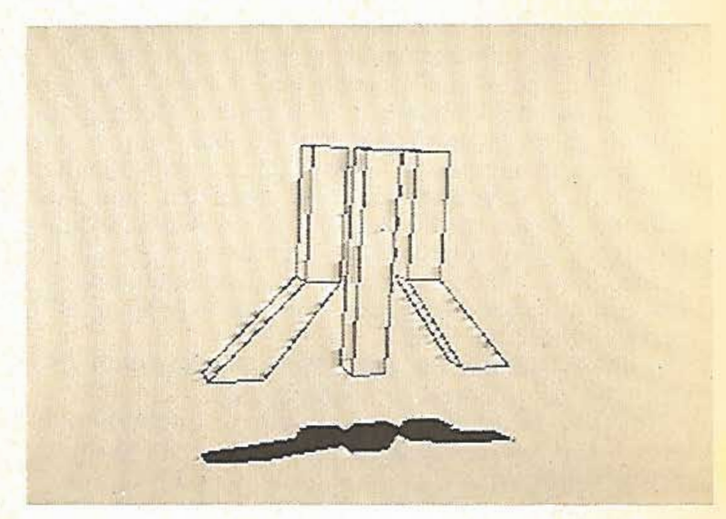


Bild 1. Ein bekanntes Zeichen in 3D

Ein nächstes Problem bereitet die Berechnung der verdeckten Kanten und Flächen (Bild 1). Über das »Hidden Line« oder »Hidden Surface«-Problem gibt es jede Menge Literatur, sogar Dissertationen und andere wissenschaftliche Arbeiten. Entsprechend groß ist auch die Anzahl der Methoden und Algorithmen zur Bewältigung dieses Problems. Die meisten Verfahren sind Scan-Algorithmen, bei denen das Bild Zeile für Zeile abgetastet, untersucht und dann entsprechend bearbeitet wird. Unsere Lösung ist ein Prioritäts-Algorithmus, der einen vertretbaren Kompromiß zwischen Aufwand und Genauigkeit darstellt. Es ist ein vereinfachter »Newell-Sancha«-Algorithmus. Zur weiteren Erklärung betrachten wir zuerst die Koordinaten-Verhältnisse.

Da wir hier dreidimensionale Objekte darstellen, braucht unser Koordinatensystem die Achsen X, Y und Z, wobei die Z-Koordinate senkrecht auf dem Bildschirm steht. Bei größer werdenden Z-Koordinaten bewegt man sich in den Bildschirm hinein.

Um verdeckte Kanten darzustellen, ist es von großer Bedeutung, wie das Objekt im Speicher des Computers abgelegt wird. Da wir von jetzt an nicht mehr nur Eckpunkte, sondern ganze Kanten des Objektes betrachten, müssen diese dem Programm als solche erkennbar sein. Zusätzlich zu diesen Informationen braucht der Computer noch Angaben über die Anzahl der Kanten jedes einzelnen Polygons und über die Verbindungslinien.

Ein Würfel besteht jetzt nicht mehr aus 8 Eckpunkten, sondern aus 6 Polygonen. Diese exakte Beschreibung des

Objektes ist eine grundlegende Bedingung, um eine Ansicht mit verdeckten Kanten zu berechnen.

Der Algorithmus funktioniert folgendermaßen: Nachdem die Rotationen und alle anderen Transformationen beendet sind, sortiert die Prozedur »SORTPOLY« alle Polygone nach der Z-Koordinate. Dadurch bekommen die Polygone eine Priorität zugewiesen. Das Polygon, das am weitesten vom Betrachter entfernt ist, erhält die niedrigste Priorität. Die Polygone werden dann in der Reihenfolge ihrer Priorität auf den Bildschirm gebracht. Zuerst wird das Polygon der niedrigsten, zuletzt das Polygon der höchsten Priorität gezeichnet. Dabei ist zu beachten, daß nicht nur die äußere Umrandung des Polygons (Linien, die das Polygon bilden) gezeichnet wird, sondern das ganze Polygon in Form einer Fläche. Im Programm erledigt das die Prozedur »DDA«. Und schon ist die 3D-Darstellung mit verdeckten Kanten fertig.

Die Sache hat aber leider einen Schönheitsfehler. Der Algorithmus funktioniert nämlich nicht immer. Es wird besonders dann kritisch, wenn sich Polygone in der Z-Ebene überlappen. Betrachten wir dazu einmal Bild 2. Wenn man nach Z-Koordinaten sortiert, erhält Polygon 2 eine höhere Priorität als Polygon 1, weil dessen Z-Wert größer ist als der von Polygon 2. Im Ergebnis würde zuerst Polygon 1, dann Polygon 2 gezeichnet werden. Von der Darstellung her ist das falsch. Dieser Fehler ließe sich durch Erweitern der Sortierkriterien zwar umgehen, was aber wieder auf Kosten der Rechenzeit gehen würde. Möchte man dennoch die Berechnungen korrekt durchführen, so ergeben sich mit Bezug auf Bild 2 folgende Sachverhalte:

Polygon 1 verdeckt Polygon 2 (im folgenden P1 und P2 genannt) nicht in dem Fall,

- daß ein einleitender Test der Z-Werte ergibt, daß sich P1 und P2 in der Z-Ebene nicht überlappen.
- daß ein Test der größten und kleinsten X- und Y-Werte der einzelnen Polygone ergibt, daß sich diese in XY-Richtung nicht überlappen.
- daß alle Kanten von P1 weiter vom Betrachter entfernt sind als die Ebene von P2. Hierzu werden die X- und Y-Werte von P1 in die Ebenengleichung (AX+BY+CZ+D=0) von P2 eingesetzt und die »Tiefe« von P2 berechnet.
- daß alle Kanten von P2 näher am Betrachter sind als die Ebene, die P1 aufspannt.
- daß ein kompletter Überlappungstest zeigt, daß sich die Polygone in X/Y-Richtung nicht überlappen.

Werden alle diese Tests durchlaufen, muß man eine Menge Geduld haben. Versuche mit vielen Objekten haben ergeben, daß man trotz ausschließlichen Sortieren nach Z-Werten fast immer die korrekte Darstellung erhält.

Die Berechnung von Schlagschatten ist gegenüber den verdeckten Kanten und Flächen eigentlich recht simpel. Grundkenntnisse der Vektorrechnung und analytischen Geometrie erleichtern das Verständnis. An der Mathematik kommt man (leider) nicht vorbei. Dazu betrachten wir nun Bild 3.

Es sei l der Vektor vom Koordinatenursprung zur punktförmigen Lichtquelle mit den Koordinaten (LX,LY,LZ) und p der Vektor zu einem beliebigen Punkt P mit den Koordinaten (PX,PY,PZ) einer beliebigen Kante eines Polygons. Dann gilt:

1) r=p-l (Vektorsubtraktion)

Wir erhalten einen neuen Vektor r mit den Koordinaten (RX,RY,RZ). Der Schlagschatten des Punktes P soll nun auf die Y-Ebene mit Y=0 fallen. Der sich ergebende Punkt soll PS heißen. Wir erhalten somit:

2) PS=l+S*r

Der Vektor vom Koordinatenursprung zum Punkt PS ergibt sich also aus dem Vektor l plus den verlängerten Vektor r. Hierzu wird der Vektor r mit einem Skalar S multipliziert und dann zum Vektor l geometrisch addiert. Da wir wissen, daß die Y-Koordinate des Punktes PS immer gleich 0 sein muß, ergibt sich für S eine lineare Gleichung in Y:

3) 0=LY+S*RY oder LY+S*RY=0 => S=-LY/RY

Bildlich gesehen ist der Schlagschatten eines Polygons eine Projektion dieses Polygons auf eine Ebene, in diesem Fall die Ebene Y=0.

Wem das alles zu theoretisch ist, der betrachte Bild 3. Dann sollte einiges klarer erscheinen.

Als letztes kann man sich überlegen, daß der Schlagschatten eines Objektes gleich der Summe der Schlagschatten seiner Polygone ist. Zeichnen wir also zunächst jedes Polygon und dann den Schlagschatten des Polygons, so erhalten wir als Ergebnis das Objekt und den Schlagschatten des Objekts.

Das abgedruckte Programm (Listing 1) ist als solches zwar in sich abgeschlossen und lauffähig, es soll aber auch eine Basis für weitere Experimente darstellen. Deshalb wurde bewußt auf aufwendige Menüs und »Kosmetik« verzichtet, weil das alles schließlich wertvollen Speicherplatz kostet. Des weiteren ist das ganze Programm in Basic geschrieben, so daß eine Adaption auf andere Computer keine großen Schwierigkeiten bereiten sollte.

Zunächst zum Hauptprogramm. Die wichtigsten Variablen und Felder werden gleich zu Beginn des Programms in den Zeilen 80 bis 120 festgelegt.

- Die Felder POLY_Z(100) und PRIO(100) enthalten Priorität und den größten Z-Wert der einzelnen Polygone, in diesem Fall für maximal 100 Einzelpolygone.
- Die Felder SX, SY, SZ, X, Y enthalten die Koordinaten der Eckpunkte der Schatten und die transformierten Koordinaten der Objekte oder Schattenpolygone. In diesem Fall darf jedes Polygon maximal 10 Eckpunkte besitzen.
- In den Zeilen 100 und 110 befinden sich die Variablen für die Bildschirmbegrenzungen. Es sind:

CLIPXL	kleinste X-Koordinate (linker Bildrand)
CLIPXR	größte X-Koordinate (rechter Bildrand)
CLIPYB	kleinste Y-Koordinate (unterer Bildrand)
CLIPYT	größte Y-Koordinate (oberer Bildrand)

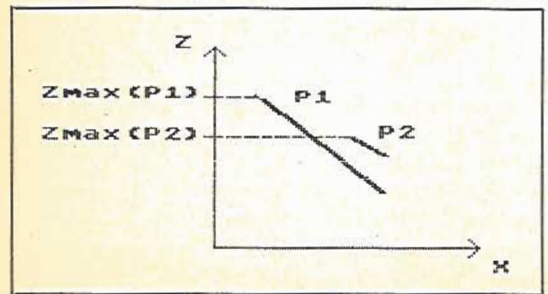


Bild 2. Fläche P1 verdeckt P2

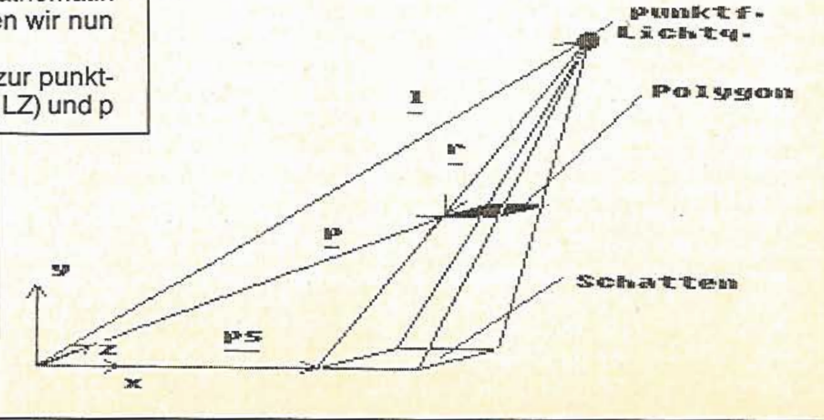


Bild 3. Berechnung und Darstellung von Schlagschatten

Im Listing ist die Einstellung eines Atari Grafikstufe 8 Screen mit 320x192 Punkten vorgegeben.

In Zeile 120 befinden sich die Koordinaten der Lichtquelle (LX,LY,LZ) und die Y-Koordinate der Schattenebene (SP).

Zur Anpassung an andere Computer ist folgendes zu beachten:

- Die Anpassung an die verschiedenen Bildschirmgrößen erfolgt mit den Variablen CLIPXL, CLIPXR, CLIPYB, CLIPYT. Diese sollten in Zeile 100 und 110 entsprechend verändert werden (zum Beispiel für Schneider oder Atari ST mit 640*400 Punkten = > CLIPXR=639, CLIPYT=399).
- Sollen mehr als 100 Polygone verwendet werden, müssen die Feldvariablen POLY_Z und PRIO entsprechend dimensioniert werden.
- Sollen mehr als 10 Eckpunkte pro Polygon verwendet werden, müssen die Feldvariablen SX, SY, SZ, X, Y entsprechend dimensioniert werden.

In letzteren beiden Fällen muß man den vorhandenen Speicherplatz berücksichtigen.

Die Transformation von Objekt- in Bildschirmkoordinaten erfolgt in der Prozedur »CALCSCREEN« in den Zeilen 800 bis 840. Die Formeln hierzu lauten:

- Zeile 810 : xc=(xv/2)*(T(0)/T(2))+(xh/2)
- Zeile 820: yc=(xv/2)*(T(1)/T(2))+(xv/2), wobei xv=maximale Y-Koordinate und xh=maximale X-Koordinate ist.

Die Transformation ist so ausgelegt, daß der Punkt 0,0 in der linken unteren Ecke liegt. Bei Computern mit dem Ursprung links oben muß der Wert yc noch einmal vom maximalen Y-Wert subtrahiert werden, um die korrekte Ansicht zu erhalten (siehe Zeile 830). Die transformierten Koordinaten befinden sich in den Variablen XC und YC.

Die diversen GRAPHICS-Befehle im Listing dienen zur Umschaltung zwischen dem Grafikmodus mit und ohne Textfenster. Auch diese Anpassung sollte keine Schwierigkeiten bereiten. Die Struktur der Prozeduren läßt sich einfach durch GOSUB und RETURN ersetzen. Gleiches gilt für die IF-, ELSE- und ENDIF-Blöcke. Probleme kann es allerdings beim UND-Operator in Zeile 3200 geben (..CA&CB...). Denn nicht jeder Computer verfügt über diese Funktion.

Zum besseren Verständnis und damit eine eventuelle Umsetzung auch gelingt, hier eine Kurzbeschreibung der einzelnen Prozeduren.

- PROC CLRMAT (480 bis 560) Löschen der Matrix MA,MB oder MC.
- PROC INITMAT (580 bis 620) Initialisieren der Matrix MA. Die Elemente der Hauptdiagonalen werden mit 1 beschrieben.
- PROC MULTMAT (640 bis 780) Multiplizieren der Matrizen MA und MB zu MC.
- PROC CALCSCREEN (800 bis 840) Transformation der dreidimensionalen in zweidimensionale Koordinaten.
- PROC CALCPOINT (860 bis 930) Multiplizieren einer Koordinate T mit der Transformationsmatrix MC.
- PROC MOVE (950 bis 1030)

Berechnen der Verschiebematrix MB zur Bewegung des Objektes. Übergabeparameter sind TX,TY und TZ.

- PROC ROTATE (1050 bis 1300)

Die Rotationsmatrix. Übergabeparameter sind ACHSE$ für die Achse und G in Gradmaß für den Winkel.

- PROC SCALE (1320 bis 1390) Skalierungsmatrix mit den Parametern SX, SY und SZ. Durch Verändern kann das Objekt in allen drei Dimensionen gestaucht oder gestreckt werden.
- PROC PERSPECTIVE (1410 bis 1460) Hier erfolgt eine perspektivische Anpassung an den Bildschirm. Durch

Verändern der Variablen wird der Fluchtpunkt verschoben. Am besten mal ausprobieren.
- PROC CLRT (1480 bis 1520)
Löschen des T-Feldes.
- PROC VIEW (1540 bis 1680)
Erstellung einer Ansicht ohne verdeckte Kanten.

Wie schon erwähnt, soll das ganze als Experimentierbasis dienen. Außerdem findet sich noch die gesondert abgedruckte Prozedur »ANIMATE« (Listing 2). Diese läßt sich mit »ENTER"D1:ANIMATE.LST"« in das Programm einbinden (Achtung: Dabei die Zeilennummern nicht verändern!). Mit Hilfe dieser Prozedur kann man Animationssequenzen zusammenstellen (Einzelbildaufnahmen).

Die Ablage der Koordinaten und Polygon-Informationen erfolgt in DATA-Zeilen. Diese werden unabhängig vom eigentlichen Programm geschrieben und dann als gesondertes File mit »LIST"D1:(NAME).IMG"« auf Diskette gespeichert. Später, nachdem das Hauptprogramm mit LOAD geladen wurde, kann das gewünschte Objekt-File mit Hilfe des ENTER-Befehls nachgeladen werden. Folgendes ist zu beachten:
- Die erste Zeilennummer ist immer 10000, und in dieser Zeile befindet sich immer die Anzahl aller Polygone des Objektes.
- Für jedes Einzelpolygon ist ein »Zehnerblock« vorgesehen, der immer bei ganzen Vielfachen von 10 beginnt (Also 10010, 10020, 10030...).
- In diesen Zeilen befindet sich dann die Anzahl der Eckpunkte des Einzelpolygons.
- Alle weiteren Zeilen enthalten die Koordinaten des Polygons. Hierbei erfolgt die Numerierung in Einserschritten.
- Zuletzt sollte noch eine Abschlußzeile eingefügt werden, die den Wert 0 enthält.

Ein einfaches quadratisches Polygon müßte so definiert werden:

10020 DATA 4: Ein Quadrat hat bekanntlich 4 Eckpunkte
10021 DATA X1,Y1,Z1: Koordinate des 1. Eckpunktes
10022 DATA X2,Y2,Z2: Koordinate des 2. Eckpunktes
10023 DATA X3,Y3,Z3: Koordinate des 3. Eckpunktes
10024 DATA X4,Y4,Z4: Koordinate des 4. Eckpunktes
10030 DATA.....(nächstes Polygon)

Auf diese Art und Weise ensteht eine bestimmte Ordnung in der Zeilennumerierung. So würde zum Beispiel die Zeilennummer 10232 entsprechend der letzten drei Ziffern besagen: 23. Polygon, 2. Eckpunkt. Hierzu finden sich auch drei Beispiele abgedruckt (Listing 3 bis 5).

Die Bedienung des Programms ist recht simpel. Folgende Kommandos stehen zur Verfügung:
MOVE, ROTATE, SCALE, VIEW und HIDE.

Die Parameter X, Y und Z beziehen sich bei den Befehlen MOVE und SCALE auf Koordinaten, beim ROTATE-Kommando sind die Winkel für jede Achse einzugeben.

Eines ist unbedingt zu beachten: Nachdem alle Bewegungs- und Rotationskommandos ausgeführt sind, muß für die korrekte Darstellung immer die Anpassungsprozedur »PERSPECTIVE« aufgerufen werden. Diese gewährleistet die korrekte Positionierung des Fluchtpunktes. Eine Kommandoeingabe kann so aussehen:
command: ROTATE (RETURN) :Der Rotationsbefehl
X,Y,Y: 10,20,0 (RETURN) :Rotiere 10 Grad über X und 20 Grad über Y
command: MOVE (RETURN) :Der Move Befehl
X,Y,Z: 0,0,10 (RETURN) :Bewegung um 10 Einheiten in Z-Richtung
command: PERSPECTIVE (RETURN) :Perspektivische Anpassung
command: VIEW :Darstellung als Drahtmodell
command: HIDE :Darstellung mit Schlagschatten und verdeckten Kanten.

(Theo Janssen/rz/hf)

```
10 REM {CTL Q}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL E}                       <LD>
20 REM {SHIFT =}        3D GRAFIK PAKET
   {SHIFT =}                            <EG>
30 REM {SHIFT =}  BY THEO JANSSEN 1986/8
7   {SHIFT =}                           <FR>
40 REM {CTL Z}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL C}                       <LE>
50 REM                                  <JS>
60 REM                                  <JT>
70 DIM MA(3,3),MB(3,3),MC(3,3)          <LA>
80 DIM P(3),T(3),POLY_Z(100),PRIO(100),S
X(10),SY(10),SZ(10)                     <PH>
90 DIM ACHSE$(1),KOM$(20),X(10),Y(10)   <XM>
100 CLIPXL=0:CLIPXR=319                 <DK>
110 CLIPYB=0:CLIPYT=191                 <AF>
120 LX=0:LY=70:LZ=0:SP=-10              <FD>
130 GRAPHICS 8:EXEC SET_COL             <HJ>
140 COLOR 1                             <XR>
150 EXEC INITMAT                        <DS>
160 INPUT "command>";KOM$               <KG>
170 IF KOM$(1,3)="VIE" THEN CLS #6:EXEC
VIEW                                    <MO>
180 IF KOM$(1,3)="INI" THEN EXEC INITMAT <MH>
190 IF KOM$(1,3)="PER" THEN EXEC PERSPEC
TIVE                                    <ZN>
200 IF KOM$(1,3)="MOV"                  <EE>
210 INPUT "X,Y,Z:>";AA,AB,AC            <IN>
220 TX=AA:TY=AB:TZ=AC                   <RT>
230 EXEC MOVE                           <DK>
240 ENDIF                               <UZ>
250 IF KOM$(1,3)="ROT"                  <GP>
260 INPUT "X,Y,Z:>";AA,AB,AC            <IX>
270 ACHSE$="X":G=AA:EXEC ROTATE         <BG>
280 ACHSE$="Y":G=AB:EXEC ROTATE         <CO>
290 ACHSE$="Z":G=AC:EXEC ROTATE         <DW>
300 ENDIF                               <US>
310 IF KOM$(1,3)="SCA"                  <IM>
320 INPUT "X,Y,Z:>";AA,AB,AC            <IQ>
330 SX=AA:SY=AB:SZ=AC                   <QP>
340 EXEC SCALE                          <FS>
350 ENDIF                               <VC>
360 IF KOM$(1,3)="HID"                  <HW>
370 GRAPHICS 24                         <XY>
380 EXEC SET_COL                        <MK>
390 EXEC HIDDENVIEW                     <YU>
400 GET KEY                             <MJ>
410 GRAPHICS 40                         <WH>
420 EXEC SET_COL                        <LZ>
430 ENDIF                               <UZ>
440 IF KOM$(1,3)="LIG" THEN INPUT "NEW L
IGHT POSITION (X,Y,Z):";LX,LY,LZ        <MC>
450 IF KOM$(1,3)="SHA" THEN INPUT "SHADO
W(S) ON Y-HEIGHT:";SP                   <WP>
460 GOTO 160                            <OJ>
470 --                                  <JT>
480 PROC CLRMAT                         <JU>
490 FOR I=0 TO 3                        <MR>
500 FOR J=0 TO 3                        <MJ>
510 IF MAT=1 THEN MA(J,I)=0             <AE>
520 IF MAT=2 THEN MB(J,I)=0             <BM>
530 IF MAT=3 THEN MC(J,I)=0             <CU>
540 NEXT J                              <GN>
550 NEXT I                              <GF>
560 ENDPROC                             <KT>
570 --                                  <JU>
580 PROC INITMAT                        <HN>
590 MAT=1:EXEC CLRMAT                   <KA>
```

```
600 MA(0,0)=1:MA(1,1)=1                    <CY>
610 MA(2,2)=1:MA(3,3)=1                    <HI>
620 ENDPROC                                <KM>
630 --                                     <JN>
640 PROC MULTMAT                           <ND>
650 MAT=3:EXEC CLRMAT                      <KL>
660 FOR K=0 TO 3                           <NF>
670 FOR J=0 TO 3                           <MY>
680 FOR I=0 TO 3                           <MR>
690 MC(J,K)=MC(J,K)+MA(I,K)*MB(J,I)        <LQ>
700 NEXT I                                 <FX>
710 NEXT J                                 <GJ>
720 NEXT K                                 <GV>
730 FOR I=0 TO 3                           <MI>
740 FOR J=0 TO 3                           <MT>
750 MA(I,J)=MC(I,J)                        <FL>
760 NEXT J                                 <GT>
770 NEXT I                                 <GL>
780 ENDPROC                                <KZ>
790 --                                     <KA>
800 PROC CALCSCREEN                        <PD>
810 XC=95.5*(T(0)/T(2))+159.5              <QY>
820 YC=95.5*(T(1)/T(2))+95.5               <ON>
830 YC=191-YC                              <EV>
840 ENDPROC                                <KS>
850 --                                     <JT>
860 PROC CALCPOINT                         <AQ>
870 FOR J=0 TO 3                           <NA>
880 FOR I=0 TO 3                           <MT>
890 T(J)=T(J)+P(I)*MC(J,I)                 <LD>
900 NEXT I                                 <FZ>
910 NEXT J                                 <GL>
920 IF T(2)<1 THEN T(2)=1                  <TF>
930 ENDPROC                                <KR>
940 --                                     <JS>
950 PROC MOVE                              <HG>
960 MAT=2:EXEC CLRMAT                      <KH>
970 MB(0,0)=1:MB(1,1)=1                    <EL>
980 MB(2,2)=1:MB(3,3)=1                    <IV>
990 MB(0,3)=TX                             <IJ>
1000 MB(1,3)=TY                            <NZ>
1010 MB(2,3)=TZ                            <PA>
1020 EXEC MULTMAT                          <ZQ>
1030 ENDPROC                               <OL>
1040 --                                    <VQ>
1050 PROC ROTATE                           <AR>
1060 DEG                                   <OO>
1070 MAT=2:EXEC CLRMAT                     <LZ>
1080 IF ACHSE$="X"                         <PG>
1090 MB(0,0)=1:MB(3,3)=1                   <DC>
1100 MB(1,1)=COS(G)                        <YQ>
1110 MB(2,1)=-SIN(G)                       <OG>
1120 MB(1,2)=SIN(G)                        <BJ>
1130 MB(2,2)=COS(G)                        <ZT>
1140 ENDIF                                 <SU>
1150 IF ACHSE$="Y"                         <PQ>
1160 MB(1,1)=1:MB(3,3)=1                   <DP>
1170 MB(0,0)=COS(G)                        <YR>
1180 MB(2,0)=SIN(G)                        <BO>
1190 MB(0,2)=-SIN(G)                       <OX>
1200 MB(2,2)=COS(G)                        <ZM>
1210 ENDIF                                 <SN>
1220 IF ACHSE$="Z"                         <QA>
1230 MB(2,2)=1:MB(3,3)=1                   <EC>
1240 MB(0,0)=COS(G)                        <YK>
1250 MB(1,0)=-SIN(G)                       <OA>
1260 MB(0,1)=SIN(G)                        <BD>
1270 MB(1,1)=COS(G)                        <ZN>
1280 ENDIF                                 <TI>
1290 EXEC MULTMAT                          <AP>
1300 ENDPROC                               <OI>
1310 --                                    <VN>
1320 PROC SCALE                            <ST>
1330 MAT=2:EXEC CLRMAT                     <LT>
1340 MB(3,3)=1                             <XE>
1350 MB(0,0)=SX                            <MB>
1360 MB(1,1)=SY                            <NN>
1370 MB(2,2)=SZ                            <OZ>
1380 EXEC MULTMAT                          <AO>
1390 ENDPROC                               <PJ>
1400 --                                    <VM>
1410 PROC PERSPECTIVE                      <SQ>
1420 MAT=2:EXEC CLRMAT                     <LS>
1430 MB(0,0)=4:MB(1,1)=4                   <EA>
1440 MB(2,2)=1:MB(3,3)=1                   <EJ>
1450 EXEC MULTMAT                          <AH>
1460 ENDPROC                               <PC>
1470 --                                    <WH>
1480 PROC CLRT                             <PX>
1490 FOR I=0 TO 3                          <XQ>
1500 T(I)=0                                <FQ>
1510 NEXT I                                <FC>
1520 ENDPROC                               <OS>
1530 --                                    <VX>
1540 PROC VIEW                             <RH>
1550 RESTORE (10000)                       <ZG>
1560 READ P_NUMBER                         <AM>
1570 FOR K=1 TO P_NUMBER                   <SP>
1580 READ E_NUMBER,XP,YP,ZP                <TS>
1590 XA=XP:YA=YP:ZA=ZP                     <IM>
1600 FOR L=2 TO E_NUMBER                   <LN>
1610 READ XL,YL,ZL                         <BF>
1620 EXEC SHOWLINE                         <VY>
1630 XP=XL:YP=YL:ZP=ZL                     <WZ>
1640 NEXT L                                <GU>
1650 XL=XA:YL=YA:ZL=ZA                     <WX>
1660 EXEC SHOWLINE                         <WK>
1670 NEXT K                                <GS>
1680 ENDPROC                               <PM>
1690 --                                    <WR>
1700 PROC SORTPOLY                         <VL>
1710 RESTORE (10000)                       <YY>
1720 READ P_NUMBER                         <AE>
1730 FOR R=1 TO P_NUMBER                   <UZ>
1740 POLY_Z(R)=0                           <CM>
1750 NEXT R                                <JN>
1760 TEXT 0,180,"sorting▲Polygons▲:"       <YU>
1770 FOR R=1 TO P_NUMBER                   <VL>
1780 TEXT 152,180,R                        <UY>
1790 ZGES=0                                <DW>
1800 READ E_NUMBER                         <VJ>
1810 FOR S=1 TO E_NUMBER                   <OA>
1820 READ X,Y,Z                            <KN>
1830 P(0)=X:P(1)=Y:P(2)=Z:P(3)=1           <WW>
1840 EXEC CLRT                             <LT>
1850 EXEC CALCPOINT                        <SN>
1860 ZGES=ZGES+T(2)                        <OO>
1870 IF T(2)>POLY_Z(R)                     <UA>
1880 POLY_Z(R)=T(2)                        <LX>
1890 ENDIF                                 <TX>
1900 NEXT S                                <JN>
1910 ZMIT=ZGES/E_NUMBER                    <DH>
1920 POLY_Z(R)=POLY_Z(R)+ZMIT              <HA>
1930 NEXT R                                <JL>
1940 FOR R=1 TO P_NUMBER                   <VG>
1950 PRIO(R)=1                             <IE>
1960 NEXT R                                <JU>
1970 FOR S=1 TO P_NUMBER                   <VZ>
1980 W=1                                   <ZW>
1990 FOR R=1 TO P_NUMBER                   <VV>
2000 IF (POLY_Z(R)>=POLY_Z(W)) THEN W=R    <AR>
2010 NEXT R                                <IO>
2020 PRIO(S)=W                             <CB>
2030 POLY_Z(W)=-9E-20                      <PR>
2040 NEXT S                                <JI>
2050 ENDPROC                               <OS>
2060 --                                    <VX>
2070 PROC HIDDENVIEW                       <AO>
2080 EXEC SORTPOLY                         <RW>
2090 CLS #6                                <EP>
2100 RESTORE (10000)                       <YK>
2110 READ P_NUMBER                         <ZQ>
2120 FOR Q=1 TO P_NUMBER                   <UB>
2130 RESTORE (10000+(PRIO(Q)*10))          <TO>
2140 READ E_NUMBER                         <VI>
2150 FOR S=1 TO E_NUMBER                   <NZ>
2160 READ X,Y,Z                            <KM>
2170 P(0)=X:P(1)=Y:P(2)=Z:P(3)=1           <WV>
2180 EXEC CLRT                             <LS>
2190 EXEC CALCPOINT                        <SM>
2200 EXEC CALCSHADOW                       <OB>
```

Listing 1. »3D-Grafik«

```
2210 EXEC CALCSCREEN                    <IR>
2220 X(S)=XC:Y(S)=YC                    <EF>
2230 NEXT S                             <JJ>
2240 FOR S=1 TO E_NUMBER                <NY>
2250 T=S+1                              <QN>
2260 IF T>E_NUMBER THEN T=1             <CM>
2270 XH=X(S):YH=Y(S)                    <YC>
2280 XJ=X(T):YJ=Y(T)                    <BB>
2290 CL=0:EXEC DDA                      <TR>
2300 NEXT S                             <JC>
2310 FOR S=1 TO E_NUMBER                <NR>
2320 T=S+1                              <QG>
2330 IF T>E_NUMBER THEN T=1             <CF>
2340 XF=X(S):YF=Y(S)                    <WD>
2350 XG=X(T):YG=Y(T)                    <YG>
2360 COLOR 1                            <YU>
2370 EXEC CLIP                          <FC>
2380 IF VIS                             <IJ>
2390 PLOT XF,YF:DRAWTO XG,YG            <VE>
2400 ENDIF                              <SP>
2410 NEXT S                             <JH>
2420 EXEC PLOTSHADOW                    <JR>
2430 NEXT Q                             <IR>
2440 ENDPROC                            <OX>
2450 --                                 <WC>
2460 PROC DDA                           <KS>
2470 LENGTH=ABS(XJ-XH)                  <UE>
2480 IF ABS(YJ-YH)>LENGTH THEN LENGTH=AB
S(YJ-YH)                                <DF>
2490 XINC=(XJ-XH)/LENGTH                <CI>
2500 YINC=(YJ-YH)/LENGTH                <CQ>
2510 XD=XH+0.5:YD=YH+0.5                <QY>
2520 FOR POINT=1 TO LENGTH              <FA>
2530 FOR N=1 TO E_NUM                   <BO>
2540 COLOR CL                           <TA>
2550 XF=XD:YF=YD:XG=X(N):YG=Y(N)        <XJ>
2560 EXEC CLIP                          <FD>
2570 IF VIS                             <IK>
2580 PLOT XF,YF                         <YM>
2590 DRAWTO XG,YG                       <NI>
2600 PLOT INT(XF),INT(YF)               <DD>
2610 DRAWTO XG,YG                       <MM>
2620 ENDIF                              <SZ>
2630 NEXT N                             <HO>
2640 XD=XD+XINC                         <DP>
2650 YD=YD+YINC                         <ET>
2660 NEXT POINT                         <QE>
2670 ENDPROC                            <PK>
2680 --                                 <WP>
2690 PROC SHOWLINE                      <AU>
2700 EXEC CLRT                          <LG>
2710 P(0)=XP:P(1)=YP:P(2)=ZP:P(3)=1     <NL>
2720 EXEC CALCPOINT                     <SD>
2730 EXEC CALCSCREEN                    <JH>
2740 XF=XC:YF=YC                        <BX>
2750 EXEC CLRT                          <LV>
2760 P(0)=XL:P(1)=YL:P(2)=ZL:P(3)=1     <EU>
2770 EXEC CALCPOINT                     <SS>
2780 EXEC CALCSCREEN                    <JW>
2790 XG=XC:YG=YC                        <DG>
2800 EXEC CLIP                          <ER>
2810 IF VIS                             <HY>
2820 PLOT XF,YF                         <YA>
2830 DRAWTO XG,YG                       <MW>
2840 ENDIF                              <TJ>
2850 ENDPROC                            <PI>
2860 --                                 <WN>
2870 PROC SET_COL                       <HO>
2880 SETCOLOR 2,0,12                    <UH>
2890 SETCOLOR 4,0,12                    <VO>
2900 SETCOLOR 1,0,0                     <FR>
2910 ENDPROC                            <OY>
2920 --                                 <WD>
2930 PROC CODE                          <CB>
2940 C=0                                <UN>
2950 IF X<CLIPXL                        <BP>
2960 C=8                                <XF>
2970 ELSE                               <UI>
2980 IF X>CLIPXR                        <GK>
2990 C=4                                <WI>
3000 ENDIF                              <SI>
3010 ENDIF                              <SL>
3020 IF Y<CLIPYB                        <VJ>
3030 C=C+2                              <HS>
3040 ELSE                               <TI>
3050 IF Y>CLIPYT                        <HO>
3060 C=C+1                              <HR>
3070 ENDIF                              <TD>
3080 ENDIF                              <TG>
3090 ENDPROC                            <PF>
3100 --                                 <VI>
3110 PROC CLIP                          <IH>
3120 VIS=1                              <AB>
3130 X=XF:Y=YF                          <XN>
3140 EXEC CODE                          <XP>
3150 CA=C                               <WI>
3160 X=XG:Y=YG                          <YT>
3170 EXEC CODE                          <XY>
3180 CB=C                               <WY>
3190 WHILE (CA<>0) OR (CB<>0)           <XQ>
3200 IF (CA&CB)<>0 THEN LET VIS=0:ENDPRO
C                                       <WT>
3210 C=CA:CZ=CA                         <KH>
3220 IF C=0 THEN LET C=CB:CZ=CB         <XX>
3230 IF (C/8)>=1                        <PJ>
3240 Y=YF+(YG-YF)*(CLIPXL-XF)/(XG-XF)   <BY>
3250 X=CLIPXL                           <PR>
3260 C=C-8                              <LF>
3270 ENDIF                              <TH>
3280 IF (C/8)>=0.5                      <EC>
3290 Y=YF+(YG-YF)*(CLIPXR-XF)/(XG-XF)   <IH>
3300 X=CLIPXR                           <SE>
3310 C=C-4                              <JE>
3320 ENDIF                              <SU>
3330 IF (C/8)>=0.25                     <OG>
3340 X=XF+(XG-XF)*(CLIPYB-YF)/(YG-YF)   <VJ>
3350 Y=CLIPYB                           <LL>
3360 C=C-2                              <IZ>
3370 ENDIF                              <TJ>
3380 IF (C/8)>=0.125                    <AS>
3390 X=XF+(XG-XF)*(CLIPYT-YF)/(YG-YF)   <NG>
3400 Y=CLIPYT                           <TY>
3410 C=C-1                              <IC>
3420 ENDIF                              <SW>
3430 IF CZ=CA                           <JZ>
3440 XF=X:YF=Y:EXEC CODE                <EF>
3450 CA=C                               <WO>
3460 ELSE                               <TW>
3470 XG=X:YG=Y:EXEC CODE                <FH>
3480 CB=C                               <XE>
3490 ENDIF                              <TR>
3500 WEND                               <TQ>
3510 ENDPROC                            <OR>
3520 --                                 <VW>
3530 PROC CALCSHADOW                    <SR>
3540 SX(S)=LX-T(0)                      <XF>
3550 SY(S)=LY-T(1)                      <YT>
3560 SZ(S)=LZ-T(2)                      <AH>
3570 SKALAR=-((LY+(-SP))/SY(S))         <IL>
3580 SX(S)=LX+SX(S)*SKALAR              <UI>
3590 SY(S)=SP                           <WF>
3600 SZ(S)=LZ+SZ(S)*SKALAR              <WG>
3610 ENDPROC                            <OT>
3620 --                                 <VY>
3630 PROC PLOTSHADOW                    <NZ>
3640 FOR S=1 TO E_NUMBER                <OH>
3650 T(0)=SX(S):T(1)=SY(S)              <IB>
3660 T(2)=SZ(S):T(3)=1                  <FY>
3670 EXEC CALCSCREEN                    <JS>
3680 X(S)=XC:Y(S)=YC                    <FG>
3690 NEXT S                             <KK>
3700 FOR S=1 TO E_NUMBER                <NX>
3710 T=S+1                              <QM>
3720 IF T>E_NUMBER THEN T=1             <CL>
3730 XH=X(S):YH=Y(S)                    <YB>
3740 XJ=X(T):YJ=Y(T)                    <BA>
3750 CL=1:EXEC DDA                      <TZ>
3760 NEXT S                             <KD>
3770 ENDPROC                            <PN>
3780 --                                 <WS>
```

Listing 1. »3D-Grafik« (Schluß)



```
455 IF KOM$(1,3)="ANI" THEN EXEC ANIMATE <AJ>
5000 PROC ANIMATE                        <KY>
5010 GRAPHICS 0:EXEC SET_COL             <VP>
5020 PRINT "STARTPOSITION"               <RS>
5030 INPUT "Winkel▲(x,y,z)";X_WS,Y_WS,Z_
WS                                       <GY>
5040 INPUT "Position▲(x,y,z)";X_PS,Y_PS,
Z_PS                                     <EF>
5050 PRINT :PRINT :PRINT "ENDPOSITION"   <OM>
5060 INPUT "Winkel▲(x,y,z)";X_WE,Y_WE,Z_
WE                                       <ZN>
5070 INPUT "Position▲(x,y,z)";X_PE,Y_PE,
Z_PE                                     <TO>
5080 PRINT :PRINT :PRINT                 <QY>
5090 INPUT "in▲welchem▲Zeitraum▲(sec.)";
TI                                       <CR>
5100 GRAPHICS 24:EXEC SET_COL            <MS>
5110 STEP=TI*25                          <AX>
5120 WMX=(X_WE-X_WS)/STEP                <ZA>
5130 WMY=(Y_WE-Y_WS)/STEP                <AM>
5140 WMZ=(Z_WE-Z_WS)/STEP                <BY>
5150 PMX=(X_PE-X_PS)/STEP                <PK>
5160 PMY=(Y_PE-Y_PS)/STEP                <QW>
5170 PMZ=(Z_PE-Z_PS)/STEP                <SI>
5180 FOR BILD=0 TO STEP                  <WZ>
5190 EXEC INITMAT                        <UG>
5200 G=X_WS:ACHSE$="X":EXEC ROTATE       <NE>
5210 G=Y_WS:ACHSE$="Y":EXEC ROTATE       <OK>
5220 G=Z_WS:ACHSE$="Z":EXEC ROTATE       <PQ>
5230 TX=X_PS:TY=Y_PS:TZ=Z_PS             <VO>
5240 EXEC MOVE                           <KZ>
5250 EXEC PERSPECTIVE                    <PC>
5260 EXEC HIDDENVIEW                     <WS>
5270 GET KEY                             <OU>
5280 X_WS=X_WS+WMX                       <PQ>
5290 Y_WS=Y_WS+WMY                       <RC>
5300 Z_WS=Z_WS+WMZ                       <RM>
5310 X_PS=X_PS+PMX                       <EY>
5320 Y_PS=Y_PS+PMY                       <GK>
5330 Z_PS=Z_PS+PMZ                       <HW>
5340 CLS #6                              <EJ>
5350 NEXT BILD                           <GX>
5360 ENDPROC                             <PE>
5370 --                                  <WJ>
```

Listing 2. »ANIMATE.LST«

```
10000 DATA▲6                             <JZ>
10010 DATA▲4                             <JF>
10011 DATA▲-1,1,-1                       <RB>
10012 DATA▲1,1,-1                        <LS>
10013 DATA▲1,-1,-1                       <RI>
10014 DATA▲-1,-1,-1                      <YU>
10020 DATA▲4                             <JJ>
10021 DATA▲1,1,1                         <IC>
10022 DATA▲-1,1,1                        <MC>
10023 DATA▲-1,-1,1                       <RS>
10024 DATA▲1,-1,1                        <MJ>
10030 DATA▲4                             <JN>
10031 DATA▲-1,1,1                        <MB>
10032 DATA▲-1,1,-1                       <RO>
10033 DATA▲-1,-1,-1                      <YX>
10034 DATA▲-1,-1,1                       <SB>
10040 DATA▲4                             <JR>
10041 DATA▲1,1,-1                        <LZ>
10042 DATA▲1,1,1                         <IP>
10043 DATA▲1,-1,1                        <MM>
10044 DATA▲1,-1,-1                       <RZ>
10050 DATA▲4                             <JV>
10051 DATA▲-1,1,1                        <MJ>
10052 DATA▲1,1,1                         <IT>
10053 DATA▲1,1,-1                        <MN>
10054 DATA▲-1,1,-1                       <SG>
10060 DATA▲4                             <JZ>
10061 DATA▲1,-1,1                        <MK>
10062 DATA▲-1,-1,1                       <SD>
10063 DATA▲-1,-1,-1                      <ZJ>
10064 DATA▲1,-1,-1                       <SH>
20000 DATA▲0                             <HG>
```

Listing 3. »WUERFEL.IMG«

```
10000 DATA▲19                            <KE>
10010 DATA▲4                             <JF>
10011 DATA▲-1,1,-1                       <RB>
10012 DATA▲1,1,-1                        <LS>
10013 DATA▲1,-1,-1                       <RI>
10014 DATA▲-1,-1,-1                      <YU>
10020 DATA▲4                             <JJ>
10021 DATA▲1,1,1                         <IC>
10022 DATA▲-1,1,1                        <MC>
10023 DATA▲-1,-1,1                       <RS>
10024 DATA▲1,-1,1                        <MJ>
10030 DATA▲4                             <JN>
10031 DATA▲-1,1,1                        <MB>
10032 DATA▲-1,1,-1                       <RO>
10033 DATA▲-1,-1,-1                      <YX>
10034 DATA▲-1,-1,1                       <SB>
10040 DATA▲4                             <JR>
10041 DATA▲1,1,-1                        <LZ>
10042 DATA▲1,1,1                         <IP>
10043 DATA▲1,-1,1                        <MM>
10044 DATA▲1,-1,-1                       <RZ>
10050 DATA▲4                             <JV>
10051 DATA▲-1,1,1                        <MJ>
10052 DATA▲1,1,1                         <IT>
10053 DATA▲1,1,-1                        <MN>
10054 DATA▲-1,1,-1                       <SG>
10060 DATA▲4                             <JZ>
10061 DATA▲1,-1,1                        <MK>
10062 DATA▲-1,-1,1                       <SD>
10063 DATA▲-1,-1,-1                      <ZJ>
10064 DATA▲1,-1,-1                       <SH>
10070 DATA▲4                             <KD>
10071 DATA▲2,0,2                         <JK>
10072 DATA▲4,0,2                         <KN>
10073 DATA▲4,0,4                         <LY>
10074 DATA▲2,0,4                         <LF>
10080 DATA▲3                             <JV>
10081 DATA▲2,0,2                         <JO>
10082 DATA▲3,2,3                         <LX>
10083 DATA▲4,0,2                         <KW>
10090 DATA▲3                             <JZ>
10091 DATA▲2,0,4                         <KY>
10092 DATA▲3,2,3                         <MB>
10093 DATA▲4,0,4                         <MG>
10100 DATA▲3                             <IS>
10101 DATA▲2,0,2                         <IL>
10102 DATA▲3,2,3                         <KU>
10103 DATA▲2,0,4                         <KB>
10110 DATA▲3                             <IW>
10111 DATA▲4,0,2                         <JN>
10112 DATA▲3,2,3                         <KY>
10113 DATA▲4,0,4                         <LD>
10120 DATA▲3                             <JA>
10121 DATA▲-2,0,-2                       <RY>
10122 DATA▲-3,2,-3                       <UM>
10123 DATA▲-4,0,-2                       <TI>
10130 DATA▲3                             <JE>
10131 DATA▲-2,0,-4                       <TM>
10132 DATA▲-3,2,-3                       <UQ>
10133 DATA▲-4,0,-4                       <UW>
10140 DATA▲3                             <JI>
10141 DATA▲-2,0,-2                       <SG>
10142 DATA▲-3,2,-3                       <UU>
10143 DATA▲-2,0,-4                       <UA>
10150 DATA▲3                             <JM>
10151 DATA▲-4,0,-2                       <TK>
```

Listing 4. »KOERPER.IMG«

```
10152 DATA -3,2,-3                  <UY>
10153 DATA -4,0,-4                  <VE>
10160 DATA 3                        <JQ>
10161 DATA -2,0,-2                  <SO>
10162 DATA -3,-2,-3                 <CJ>
10163 DATA -4,0,-2                  <TY>
10170 DATA 3                        <JU>
10171 DATA -2,0,-4                  <UC>
10172 DATA -3,-2,-3                 <CN>
10173 DATA -4,0,-4                  <VM>
10180 DATA 3                        <JY>
10181 DATA -2,0,-2                  <SW>
10182 DATA -3,-2,-3                 <CR>
10183 DATA -2,0,-4                  <UQ>
10190 DATA 3                        <KC>
10191 DATA -4,0,-2                  <UA>
10192 DATA -3,-2,-3                 <CV>
10193 DATA -4,0,-4                  <VU>
20000 DATA 0                        <HG>
```

Listing 4. »KOERPER.IMG« (Schluß)

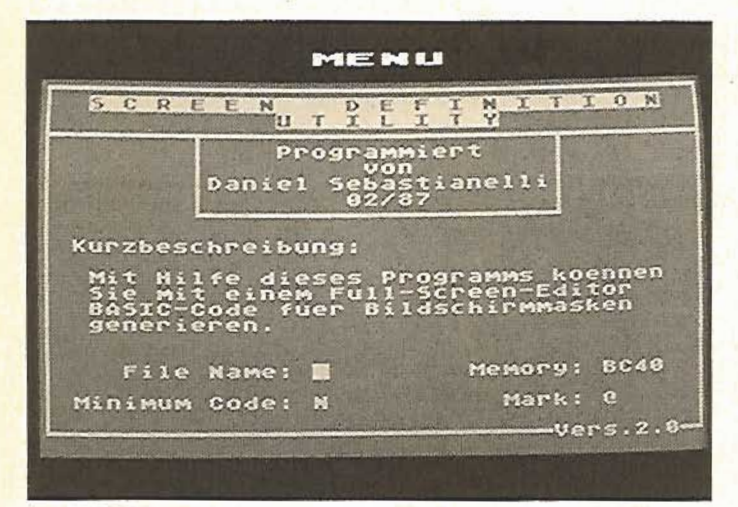
```
10000 DATA 26                       <JD>
10010 DATA 4                        <JF>
10011 DATA -.2,-1.4,-.2             <AK>
10012 DATA -.2,1.4,-.2              <MD>
10013 DATA .2,1.4,-.2               <ZK>
10014 DATA .2,-1.4,-.2              <MI>
10020 DATA 4                        <JJ>
10021 DATA -.2,-1.4,.2              <MN>
10022 DATA -.2,1.4,.2               <ZZ>
10023 DATA .2,1.4,.2                <OZ>
10024 DATA .2,-1.4,.2               <AE>
10030 DATA 4                        <JN>
10031 DATA -.2,-1.4,-.2             <AS>
10032 DATA .2,-1.4,-.2              <MG>
10033 DATA .2,-1.4,.2               <AD>
10034 DATA -.2,-1.4,.2              <NG>
10040 DATA 4                        <JR>
10041 DATA -.2,1.4,-.2              <MK>
10042 DATA .2,1.4,-.2               <ZR>
10043 DATA .2,1.4,.2                <PH>
10044 DATA -.2,1.4,.2               <AR>
10050 DATA 4                        <JV>
10051 DATA -.2,-1.4,.2              <MZ>
10052 DATA -.2,1.4,.2               <AL>
10053 DATA -.2,1.4,-.2              <MY>
10054 DATA -.2,-1.4,-.2             <BP>
10060 DATA 4                        <JZ>
10061 DATA .2,-1.4,.2               <AF>
10062 DATA .2,1.4,.2                <PK>
10063 DATA .2,1.4,-.2               <AE>
10064 DATA .2,-1.4,-.2              <NC>
10070 DATA 4                        <KD>
10071 DATA -.8,1.4,-.2              <QC>
10072 DATA -.4,1.4,-.2              <OD>
10073 DATA -.4,-.2,-.2              <KM>
10074 DATA -.8,-.2,-.2              <MV>
10080 DATA 4                        <KH>
10081 DATA -.8,1.4,.2               <DY>
10082 DATA -.4,1.4,.2               <BZ>
10083 DATA -.4,-.2,.2               <YI>
10084 DATA -.8,-.2,.2               <AR>
10090 DATA 4                        <KL>
10091 DATA -.8,1.4,-.2              <QK>
10092 DATA -.8,1.4,.2               <EH>
10093 DATA -.4,1.4,.2               <CI>
10094 DATA -.4,1.4,-.2              <OV>
10100 DATA 4                        <JE>
10101 DATA -.4,1.4,-.2              <MZ>
10102 DATA -.4,1.4,.2               <AW>
10103 DATA -.4,-.2,.2               <XF>
10104 DATA -.4,-.2,-.2              <JS>
10110 DATA 4                        <JI>
10111 DATA -.8,1.4,-.2              <PH>
10112 DATA -.8,1.4,.2               <DE>
10113 DATA -.8,-.2,.2               <ZW>
10114 DATA -.8,-.2,-.2              <MA>
10120 DATA 4                        <JM>
10121 DATA .8,1.4,-.2               <CH>
10122 DATA .4,1.4,-.2               <AM>
10123 DATA .4,-.2,-.2               <XB>
10124 DATA .8,-.2,-.2               <ZG>
10130 DATA 4                        <JQ>
10131 DATA .8,1.4,.2                <RW>
10132 DATA .4,1.4,.2                <QB>
10133 DATA .4,-.2,.2                <MQ>
10134 DATA .8,-.2,.2                <OV>
10140 DATA 4                        <JU>
10141 DATA .8,1.4,-.2               <CP>
10142 DATA .8,1.4,.2                <SF>
10143 DATA .4,1.4,.2                <QK>
10144 DATA .4,1.4,-.2               <BE>
10150 DATA 4                        <JY>
10151 DATA .4,1.4,-.2               <AT>
10152 DATA .4,1.4,.2                <QJ>
10153 DATA .4,-.2,.2                <MY>
10154 DATA .4,-.2,-.2               <XS>
10160 DATA 4                        <KC>
10161 DATA .8,1.4,-.2               <CX>
10162 DATA .8,1.4,.2                <SN>
10163 DATA .8,-.2,.2                <PC>
10164 DATA .8,-.2,-.2               <ZW>
10170 DATA 4                        <KG>
10171 DATA -.8,-.2,-.2              <MJ>
10172 DATA -.4,-.2,-.2              <KK>
10173 DATA -1.2,-1.4,-.2            <UF>
10174 DATA -1.8,-1.4,-.2            <XW>
10180 DATA 4                        <KK>
10181 DATA -.8,-.2,.2               <AF>
10182 DATA -.4,-.2,.2               <YG>
10183 DATA -1.2,-1.4,.2             <EP>
10184 DATA -1.8,-1.4,.2             <IG>
10190 DATA 4                        <KO>
10191 DATA -.8,-.2,.2               <AJ>
10192 DATA -1.8,-1.4,.2             <IA>
10193 DATA -1.8,-1.4,-.2            <XZ>
10194 DATA -.8,-.2,-.2              <NG>
10200 DATA 4                        <JH>
10201 DATA -.4,-.2,.2               <WY>
10202 DATA -1.2,-1.4,.2             <DH>
10203 DATA -1.2,-1.4,-.2            <TG>
10204 DATA -.4,-.2,-.2              <JV>
10210 DATA 4                        <JL>
10211 DATA -1.8,-1.4,-.2            <WM>
10212 DATA -1.2,-1.4,-.2            <TF>
10213 DATA -1.2,-1.4,.2             <DQ>
10214 DATA -1.8,-1.4,.2             <HH>
10220 DATA 4                        <JP>
10221 DATA .8,-.2,-.2               <YU>
10222 DATA .4,-.2,-.2               <WZ>
10223 DATA 1.2,-1.4,-.2             <DA>
10224 DATA 1.8,-1.4,-.2             <GL>
10230 DATA 4                        <JT>
10231 DATA .8,-.2,.2                <OJ>
10232 DATA .4,-.2,.2                <MO>
10233 DATA 1.2,-1.4,.2              <PD>
10234 DATA 1.8,-1.4,.2              <SO>
10240 DATA 4                        <JX>
10241 DATA .8,-.2,.2                <ON>
10242 DATA 1.8,-1.4,.2              <SI>
10243 DATA 1.8,-1.4,-.2             <GO>
10244 DATA .8,-.2,-.2               <ZR>
10250 DATA 4                        <KB>
10251 DATA .4,-.2,.2                <MR>
10252 DATA 1.2,-1.4,.2              <PG>
10253 DATA 1.2,-1.4,-.2             <DM>
10254 DATA .4,-.2,-.2               <XV>
10260 DATA 4                        <KF>
10261 DATA 1.8,-1.4,-.2             <GM>
10262 DATA 1.2,-1.4,-.2             <DL>
10263 DATA 1.2,-1.4,.2              <PP>
10264 DATA 1.8,-1.4,.2              <TA>
20000 DATA 0                        <HG>
```

Listing 5. »LOGO.IMG«

Benutzerfreundliche Programme

Um ein Programm benutzerfreundlich zu gestalten, bedarf es viel Zeit. Zeit, die eigentlich besser genutzt werden könnte. Mit unserem Programm sparen Sie diese Zeit, denn Menüsteuerungen sind damit spielend leicht programmierbar.

Haben Sie schon einmal versucht, ein Programm möglichst benutzerfreundlich zu gestalten? Normalerweise geschieht dies über Menüs, die vom Anwender abgerufen werden und ihm die jeweilige aufgerufene Funktion erklären. Schön und gut. Nur, bis man sein Programm mit einem Menü versehen hat, vergeht meistens nochmal die gleiche Zeit wie bei der Entwicklung des gesamten Programms.



Das »Screen-Definition-Utility« läßt sich in Programme einbinden und erzeugt Bildschirmmasken mit allen Abfragen

Um Ihnen die Arbeit bei der Gestaltung von Menüs zu erleichtern, haben wir das »Screen-Definition-Utility« veröffentlicht, mit dem sich einfach und komfortabel eine komplette Bildschirmmaske aufbauen läßt. In diese lassen sich an beliebiger Stelle Abfragen nach Zahlen oder Zeichenketten einbauen. Anschließend schreibt unser Programm verschiedene Files auf Diskette, die in eigene Programme einbindbar sind und die die Bildschirmmaske mit allen Abfragen erzeugen.

»SDU« für meisterhafte Menüs

Das Programm ist in Turbo-Basic geschrieben (Listing 1). Es benötigt noch ein weiteres File. Dieses ist als AMPEL-Listing abgedruckt (Listing 2) und muß auf der gleichen Diskette wie das Screen-Definition-Utility unter dem Namen »SDU« vorhanden sein.

Nach dem Starten des Programms werden Sie nach dem Filenamen gefragt, unter dem die einzelnen von Screen-Definition-Utility erzeugten Files abgelegt werden. Geben Sie bitte keine Extension an, da das Programm mehrere Files mit dem angegebenen Namen erzeugt, zur Unterscheidung aber jeweils eine andere Extension einsetzt.

Anschließend fragt das Programm, ob das erzeugte Basic-Programm in kurzer oder langer Form gespeichert werden soll. Bei der langen Form wird aus jeder Zeichenfolge, die zwischen zwei Leerzeichen steht, ein eigener Print-Befehl erzeugt, die kurze Fassung verwandelt jede 40-Zeichen-Zeile vom Bildschirm in einen Print-Befehl.

Memory fragt nach der hexadezimalen Adresse, ab der Screen-Definition-Utility ein Maschinensprache-Programm erzeugt, das dann in eigenen Programme eingebunden wird. Dieses Maschinenprogramm lädt eine vorgefertigte Eingabemaske von der Diskette. Die Adresse muß zwischen $9000 und $CC40 liegen. Drücken Sie hier einfach nur <RETURN>, wird die Adresse $BC40 angenommen.

Schließlich folgt die Frage nach dem **Mark**-Zeichen. Mit diesem Zeichen markieren Sie in der Eingabemaske Stellen, an der Ihr späteres Programm eine Eingabe fordern soll. Dieses Zeichen ist normalerweise das <@>-Zeichen (Klammeraffe, Affenschaukel, wie auch immer), Sie können jedoch jedes beliebige andere Zeichen dafür definieren.

Beliebige Abfragepositionen

Nach dieser Eingabe gelangen Sie in den Editor, mit dem sich Eingabemasken konstruieren lassen. Falls auf der Diskette schon eine Maske unter dem im Eingangsmenü eingegebenen Namen vorhanden ist, wird diese automatisch geladen. Mit dem Cursor lassen sich an verschiedenen Stellen Zeichen setzen. Versehentlich gesetzte Zeichen löschen Sie wieder mit der <SPACE>-Taste. Eine ganze Zeile wird mit der Tastenkombination <SHIFT> und <DELETE> gelöscht, und folgende Zeilen werden nachgerückt. Die Bildschirmposition in der rechten untersten Ecke kann nicht belegt werden, da der Bildschirminhalt sonst eine Zeile nach oben rücken würde und damit der Bildschirmaufbau zerstört würde. Mit <ESC> verläßt man den Editor. Steht der Cursor beim Drücken von <ESC> in der obersten linken Ecke, werden die folgenden beiden Programmmteile nicht ausgeführt, und man befindet sich sofort im »Order«-Programm.

Steht der Cursor jedoch beim Verlassen des Editors an einer anderen Position, wird die Maske auf der Diskette gespeichert und anschließend die »Searching«-Routine aufgerufen. Diese durchsucht die Maske nach im Editor eingegebenen Mark-Zeichen. Findet die Search-Routine ein Mark, so fragt sie nach dem Variablennamen, den sie der an dieser Stelle eingebenen Zeichenfolge oder Zahl geben soll.

Sind alle Marks gefunden und der ganze Bildschirm durchsucht, wird die **Order**-Routine aufgerufen. Mit dieser legt man die Reihenfolge fest, in der die einzelnen Abfragen auf dem Bildschirm erscheinen sollen. Drücken Sie gleich zu Anfang <ESC> und wurde vorher die Searching-Routine nicht aufgerufen, so wird die Einstellung von einer vorher vorhandenen Maske übernommen und gleich der nächste Programmteil angesprungen. Ansonsten steht der Cursor auf der ersten Eingabe. Wollen Sie, daß diese Marke nicht als erstes in Ihrem Programm

abgefragt wird, springen Sie mit <+> oder <*> die nächste Marke an. Mit <RETURN> wird dann die Marke als nächstes in der Abfragereihenfolge festgelegt. Mit <ESC> löschen Sie eine komplette Abfragereihenfolge.

Ist die komplette Abfragereihenfolge der Marken gespeichert, so springt das Programm schließlich in das **Generate**-Menü. Hier werden Sie nach der ersten Zeilennummer gefragt, unter der die Eingabemaske in Ihrem Programm stehen soll, sowie nach dem Abstand der Zeilennummern. Anschließend wird das Eingabemasken-Programm auf Diskette gespeichert.

Auf Ihrer Diskette stehen dann insgesamt sechs Files, die das Eingabemaskenprogramm benötigt. Um die Maske in Ihr Programm einzubinden, geben Sie die Befehle »ENTER "D:Filename.BAS"« sowie ENTER "D:Filename.INP"« ein. In Ihrem Programm muß dann am Anfang noch der Befehl »BLOAD "D:Filename.ASM"« vorkommen. Jetzt wird die definierte Eingabemaske ohne Probleme geladen und ausgeführt.

(Daniel Sebastianelli/hf)

Steckbrief	
Programm:	Screen-Definition-Utility
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsumme und AMPEL
Datenträger:	Diskette

```
10 POKE 106,191                                  <KO>
20 DIM PIC_NAME$(15),BAS_NAME$(15),MEM_N
AME$(15),EXT_NAME$(15),INP_NAME$(15),ASM
_NAME$(15),NAME$(11)                             <WG>
30 DIM Z(88,43),A$(40),KEY$(41),KEY1$(12
)                                                <ZA>
40 KEY$="$(,)0123456789ABCDEFGHIJKLMNOPQ
RSTUVWXYZ_"                                      <LR>
50 KEY1$="{ESC DEL}{ESC CTL <}{ESC CTL >
}{ESC SHIFT >}{ESC CTL DEL}{ESC SHIFT DE
L}{ESC CTL +}{ESC CTL *}{ESC CTL -}{ESC
CTL =}{ESC ESC}▲"                                <JQ>
60 POKE 82,0                                     <GC>
70 FOR I=0 TO 8                                  <GK>
80 READ A$                                       <KY>
90 FOR J=0 TO 19                                 <LV>
100 POKE $0600+J+I*20,ASC(A$(J+1))               <BH>
110 NEXT J                                       <GD>
120 NEXT I                                       <FV>
130 GRAPHICS 0                                   <SM>
140 DL=DPEEK(560)                                <AK>
150 POKE DL-2,112                                <JL>
160 POKE DL-1,70                                 <FE>
170 POKE DL+2,112                                <IR>
180 DPOKE 560,DL-2                               <JV>
190 BILD=DPEEK(88)                               <IB>
200 # MENUE                                      <FB>
210 CLS                                          <AU>
220 POKE 752,0                                   <IW>
230 POKE 729,40                                  <OT>
240 POKE 730,5                                   <JY>
250 DPOKE DL,$0600+160                           <SR>
260 CLOSE #1                                     <LK>
270 POSITION 16,18                               <DB>
280 PUT 158                                      <PF>
290 I=LEN(NAME$)+1                               <LR>
300 TRAP #ERROR                                  <DT>
310 OPEN #1,4,0,"D:SDU"                          <WY>
320 BGET #1,BILD,959                             <WE>
330 CLOSE #1                                     <LF>
340 PUT 158                                      <OY>
350 NAME$="D:"                                   <JM>
360 I=2                                          <GE>
370 REPEAT                                       <DM>
380 I=I+1                                        <QS>
390 REPEAT                                       <DQ>
400 # INPUT_NAME                                 <KP>
410 POKE 702,64                                  <NZ>
420 POKE 694,0                                   <LE>
430 TRAP #INPUT_NAME:GET KEY                     <JV>
440 IF I=3                                       <OM>
450 OK=INSTR(KEY$,CHR$(KEY),14)                  <ZS>
460 ELSE                                         <YC>
470 OK=INSTR(KEY$,CHR$(KEY),4)                   <TQ>
480 ENDIF                                        <VJ>
490 IF KEY=126 AND I>3                           <WR>
500 I=I-1                                        <QU>
510 PUT KEY                                      <RX>
520 ENDIF                                        <UY>
530 UNTIL OK>0 OR KEY=155 AND I>3                <BE>
540 PUT KEY                                      <SD>
550 IF KEY<>155 THEN NAME$(I)=CHR$(KEY)          <PG>
560 UNTIL I=10 OR KEY=155                        <HO>
570 NAME$=NAME$(1,I-1)                           <NK>
580 POSITION 16,20                               <YJ>
590 PUT 78                                       <UY>
600 REPEAT                                       <DB>
610 # INPUT_MIN_CODE                             <YK>
620 POKE 702,64                                  <OD>
630 POKE 694,0                                   <LI>
640 TRAP #INPUT_MIN_CODE:GET MIN_CODE            <GZ>
650 UNTIL MIN_CODE=74 OR MIN_CODE=78 OR
MIN_CODE=155                                     <XI>
660 IF MIN_CODE=155 THEN MIN_CODE=78             <KP>
670 POSITION 16,20                               <YI>
680 PUT MIN_CODE                                 <JQ>
690 MEMORY=48192                                 <LE>
700 A$="BC40"                                    <RT>
710 POSITION 33,18                               <CD>
720 ? A$                                         <MN>
730 POSITION 33,18                               <CH>
740 PUT 158                                      <PC>
750 I=0                                          <FS>
760 REPEAT                                       <DO>
770 I=I+1                                        <QU>
780 REPEAT                                       <DS>
790 # INPUT_MEMORY                               <EL>
800 POKE 702,64                                  <OB>
810 POKE 694,0                                   <LG>
820 TRAP #INPUT_MEMORY:GET KEY                   <VI>
830 HEX=INSTR(KEY$(5,20),CHR$(KEY))              <CH>
840 IF KEY=126 AND I>1                           <UT>
850 I=I-1                                        <RH>
860 PUT KEY                                      <SK>
870 ELSE                                         <YI>
880 IF I=1                                       <OE>
890 ? "▲▲▲▲"                                     <ED>
900 POSITION 33,18                               <CD>
910 PUT 158                                      <OY>
920 ENDIF                                        <VC>
930 ENDIF                                        <VE>
940 UNTIL HEX>0 OR KEY=155 AND I=1               <FR>
950 PUT KEY                                      <SJ>
960 A$(I)=CHR$(KEY)                              <UK>
970 UNTIL KEY=155 AND I=1 OR I=4                 <OB>
980 IF I=4                                       <PJ>
990 MEMORY=DEC(A$(1,4))                          <ID>
1000 MEMORY=(MEMORY+36864+ABS(MEMORY-368
64))/2                                           <SL>
1010 MEMORY=(MEMORY+52288-ABS(MEMORY-522
88))/2                                           <QM>
1020 ENDIF                                       <SM>
1030 POSITION 33,18                              <TW>
1040 ? HEX$(MEMORY)                              <HT>
1050 MARK=64                                     <AA>
1060 POSITION 33,20                              <PB>
1070 PUT MARK                                    <CF>
1080 REPEAT                                      <EX>
1090 REPEAT                                      <FA>
1100 # INPUT_MARK                                <GJ>
1110 POKE 702,64                                 <VD>
1120 POKE 694,0                                  <QN>
```

```
1130 TRAP #INPUT_MARK:GET KEY                <OG>
1140 OK=INSTR(KEY$(5,40),CHR$(KEY))          <XP>
1150 OK1=INSTR(KEY1$,CHR$(KEY))              <IV>
1160 UNTIL OK=0 AND OK1=0 OR KEY=155         <AQ>
1170 IF KEY<>155 THEN MARK=KEY               <UC>
1180 POSITION 33,20                          <PJ>
1190 PUT MARK                                <CN>
1200 UNTIL KEY=155                           <XB>
1210 MARK=PEEK(BILD+833)                     <RP>
1220 POKE 729,10                             <UK>
1230 POKE 730,1                              <MY>
1240 PUT 158                                 <OR>
1250 CLS                                     <TY>
1260 I=LEN(NAME$)+1                          <BH>
1270 MEM_NAME$=NAME$                         <CP>
1280 MEM_NAME$(I)=".MEM"                     <GY>
1290 PIC_NAME$=NAME$                         <BN>
1300 PIC_NAME$(I)=".PIC"                     <BT>
1310 BAS_NAME$=NAME$                         <AF>
1320 BAS_NAME$(I)=".BAS"                     <XP>
1330 EXT_NAME$=NAME$                         <HQ>
1340 EXT_NAME$(I)=".EXT"                     <BU>
1350 INP_NAME$=NAME$                         <EW>
1360 INP_NAME$(I)=".INP"                     <QG>
1370 ASM_NAME$=NAME$                         <DR>
1380 ASM_NAME$(I)=".ASM"                     <KE>
1390 TRAP #EMPTY                             <UW>
1400 OPEN #1,4,0,PIC_NAME$                   <YI>
1410 BGET #1,BILD,959                        <PH>
1420 POKE 93,PEEK(BILD)                      <FN>
1430 # EMPTY                                 <PH>
1440 CLOSE #1                                <ND>
1450 PUT 158                                 <OY>
1460 DPOKE DL,$0600                          <KG>
1470 REPEAT                                  <FC>
1480 REPEAT                                  <FF>
1490 # INPUT_CHAR                            <AK>
1500 TRAP #INPUT_CHAR:GET KEY                <EZ>
1510 UNTIL (PEEK(84)<>23 OR KEY<>155) AN
D KEY<>125                                   <XH>
1520 IF PEEK(84)=23 AND PEEK(85)=39 THEN
 PUT 31                                      <WC>
1530 POKE 99,119                             <AO>
1540 PUT KEY                                 <UX>
1550 UNTIL KEY=27                            <PC>
1560 IF PEEK(84)=0 AND PEEK(85)=0            <GS>
1570 TRAP #MENUE                             <HR>
1580 OPEN #1,4,0,MEM_NAME$                   <CA>
1590 GET #1;X                                <VK>
1600 FOR I=0 TO X                            <UY>
1610 FOR J=0 TO 2                            <WP>
1620 GET #1;Z                                <VR>
1630 IF J=1                                  <KV>
1640 GET #1;ZH                               <IR>
1650 Z=ZH*256+Z                              <TJ>
1660 ENDIF                                   <TK>
1670 Z(I,J)=Z                                <GD>
1680 NEXT J                                  <GK>
1690 J=J-1                                   <MW>
1700 REPEAT                                  <EN>
1710 J=J+1                                   <LI>
1720 GET #1;Z                                <VT>
1730 Z(I,J)=Z                                <FT>
1740 UNTIL Z=0                               <YB>
1750 NEXT I                                  <FS>
1760 MFLAG=1                                 <CD>
1770 CLOSE #1                                <NS>
1780 TRAP #ERROR                             <RL>
1790 OPEN #1,4,0,PIC_NAME$                   <ZP>
1800 BGET #1,BILD,959                        <PM>
1810 CLOSE #1                                <NC>
1820 ELSE                                    <TQ>
1830 POKE 752,1:PUT 27                       <XC>
1840 TRAP #ERROR                             <RB>
1850 OPEN #1,8,0,PIC_NAME$                   <BJ>
1860 BPUT #1,BILD,959                        <XN>
1870 CLOSE #1                                <NU>
1880 DPOKE DL,$0600+40                       <WM>
1890 IM_BILD=BILD-1                          <FY>
1900 X=-1                                    <OX>
1910 Y=1                                     <ZN>
1920 REPEAT                                  <EX>
1930 IM_BILD=IM_BILD+1                       <HK>
1940 CHR=PEEK(IM_BILD)                       <GP>
1950 INV_CHR=(CHR+128) MOD 256               <BQ>
1960 POKE IM_BILD,INV_CHR                    <WC>
1970 IF CHR<>91 AND (CHR>0 OR MIN_CODE=7
4)                                           <QO>
1980 IF CHR=MARK                             <ON>
1990 DPOKE DL,$0600+20                       <VB>
2000 CHR=0                                   <UJ>
2010 I=5                                     <WK>
2020 Y=1                                     <YZ>
2030 X=X+1                                   <SP>
2040 Z(X,Y)=IM_BILD-BILD                     <BW>
2050 Y=Y+1                                   <TJ>
2060 Z(X,Y)=0                                <UZ>
2070 Y=Y+1                                   <TP>
2080 REPEAT                                  <EY>
2090 REPEAT                                  <FB>
2100 # INPUT_VAR                             <OF>
2110 POKE 702,64                             <VE>
2120 POKE 694,0                              <QO>
2130 TRAP #INPUT_VAR:GET KEY                 <QX>
2140 IF I=5                                  <LY>
2150 OK=INSTR(KEY$,CHR$(KEY),14)             <WW>
2160 ELSE                                    <TP>
2170 OK=INSTR(KEY$,CHR$(KEY))                <GT>
2180 ENDIF                                   <TH>
2190 IF KEY=126 AND I>5                      <XB>
2200 I=I-1                                   <LA>
2210 POKE $0600+20+I,0                       <EA>
2220 ENDIF                                   <SR>
2230 UNTIL OK>0 OR KEY=155                   <GD>
2240 IF KEY<>126 AND KEY<>155                <OR>
2250 POKE $0600+20+I,KEY-32                  <ES>
2260 I=I+1                                   <LA>
2270 ENDIF                                   <TG>
2280 UNTIL KEY=155 OR I=19                   <KM>
2290 DPOKE DL,$0600+40                       <WE>
2300 FOR J=5 TO I-1                          <CP>
2310 Z(X,Y)=PEEK($0600+20+J)+32              <VP>
2320 Y=Y+1                                   <TG>
2330 POKE $0600+20+J,0                       <FC>
2340 NEXT J                                  <FT>
2350 Y=1                                     <ZO>
2360 ELSE                                    <TT>
2370 IF Y=1                                  <QH>
2380 X=X+1                                   <TK>
2390 Z(X,Y)=IM_BILD-BILD                     <CR>
2400 Y=Y+1                                   <TC>
2410 Z(X,Y)=1                                <VF>
2420 Y=Y+1                                   <TI>
2430 ENDIF                                   <SY>
2440 IF CHR=64 THEN CHR=0                    <WK>
2450 IF CHR<64                               <FK>
2460 ASC_CHR=CHR+32                          <BE>
2470 ELSE                                    <TY>
2480 IF CHR>63 AND CHR<96                    <CT>
2490 ASC_CHR=CHR-64                          <GJ>
2500 ELSE                                    <TF>
2510 IF CHR>127 AND CHR<192                  <ZP>
2520 ASC_CHR=CHR+32                          <AU>
2530 ELSE                                    <TO>
2540 IF CHR>191 AND CHR<224                  <WH>
2550 ASC_CHR=CHR-64                          <FZ>
2560 ELSE                                    <TX>
2570 ASC_CHR=CHR                             <FK>
2580 ENDIF                                   <TP>
2590 ENDIF                                   <TS>
2600 ENDIF                                   <ST>
2610 ENDIF                                   <SW>
2620 Z(X,Y)=ASC_CHR                          <MW>
2630 Y=Y+1                                   <TP>
2640 IF (IM_BILD-BILD+1) MOD 40=0 THEN Y
=1                                           <FZ>
```

Listing 1. »Screen-Definition-Utility«. Bitte mit Prüfsummer eingeben.

```
2650 ENDIF                                  <TI>
2660 ELSE                                   <TZ>
2670 Y=1                                    <AA>
2680 ENDIF                                  <TR>
2690 POKE IM_BILD,CHR                       <LB>
2700 IF X>87                                <SE>
2710 GRAPHICS 0                             <BL>
2720 PUT 25                                 <PK>
2730 ? "ERROR-▲MEHR▲ALS▲88▲FELDER"          <JU>
2740 ? "VERSUCHE▲MINIMUM-CODE=""J"""        <BR>
2750 END                                    <QT>
2760 ENDIF                                  <TN>
2770 UNTIL CHR=91                           <JY>
2780 IF X=-1 THEN GO# MENUE                 <HD>
2790 POKE IM_BILD,0                         <FQ>
2800 TRAP #ERROR                            <QQ>
2810 OPEN #1,8,0,EXT_NAME$                  <RJ>
2820 BPUT #1,BILD,959                       <XC>
2830 CLOSE #1                               <NJ>
2840 TRAP #ERROR                            <RC>
2850 OPEN #1,8,0,ASM_NAME$                  <GA>
2860 PUT #1,255                             <IX>
2870 PUT #1,255                             <JA>
2880 PUT #1,MEMORY MOD 256                  <UI>
2890 PUT #1,MEMORY DIV 256                  <XX>
2900 PUT #1,(MEMORY+958) MOD 256            <HS>
2910 PUT #1,(MEMORY+958) DIV 256            <LZ>
2920 BPUT #1,BILD,959                       <XE>
2930 CLOSE #1                               <NL>
2940 ENDIF                                  <TL>
2950 DPOKE DL,$0600+60                      <XW>
2960 REPEAT                                 <FK>
2970 POS=Z(Z,1)+BILD                        <LS>
2980 IF Z(Z,2)=1                            <DF>
2990 CHR=PEEK(POS)                          <TX>
3000 ELSE                                   <SW>
3010 CHR=31                                 <QK>
3020 ENDIF                                  <SO>
3030 INV_CHR=(CHR+128) MOD 256              <AU>
3040 POKE POS,INV_CHR                       <SK>
3050 TRAP #ERROR                            <QQ>
3060 GET KEY                                <OL>
3070 IF KEY=27                              <KA>
3080 IF MFLAG=1                             <XO>
3090 EXIT                                   <ZG>
3100 ELSE                                   <SY>
3110 FOR I=0 TO X                           <UT>
3120 Z(I,0)=0                               <AG>
3130 NEXT I                                 <FC>
3140 L=0                                    <WA>
3150 ENDIF                                  <SZ>
3160 ELSE                                   <TQ>
3170 IF MFLAG=1                             <XN>
3180 FOR I=0 TO X                           <VO>
3190 Z(I,0)=0                               <BB>
3200 NEXT I                                 <EV>
3210 L=0                                    <VT>
3220 MFLAG=0                                <AX>
3230 ENDIF                                  <SV>
3240 IF KEY=42                              <ID>
3250 # RIGHT                                <HF>
3260 REPEAT                                 <EX>
3270 Z=Z+1                                  <UI>
3280 IF Z>X THEN Z=0                        <RR>
3290 UNTIL Z(Z,0)=0 OR L-1=X                <AD>
3300 ELSE                                   <TC>
3310 IF KEY=43                              <IK>
3320 REPEAT                                 <EN>
3330 Z=Z-1                                  <UQ>
3340 IF Z<0 THEN Z=X                        <EJ>
3350 UNTIL Z(Z,0)=0                         <NC>
3360 ELSE                                   <TU>
3370 IF KEY=155                             <NG>
3380 L=L+1                                  <MZ>
3390 Z(Z,0)=L                               <UL>
3400 GO# RIGHT                              <AZ>
3410 ENDIF                                  <ST>
3420 ENDIF                                  <SW>
3430 ENDIF                                  <SZ>
3440 ENDIF                                  <TC>
3450 POKE POS,CHR                           <GS>
3460 UNTIL L-1=X                            <VO>
3470 PUT 158                                <PG>
3480 CLS                                    <UN>
3490 POKE 752,0:PUT 158                     <SA>
3500 DPOKE DL,$0600+100                     <JL>
3510 POSITION 13,8                          <IW>
3520 ? "Line▲Number:";                      <WS>
3530 POSITION 16,10                         <PE>
3540 ? "Increase:";                         <LC>
3550 REPEAT                                 <FA>
3560 POSITION 26,8                          <LW>
3570 ? "10",                                <ZR>
3580 POSITION 26,10                         <QI>
3590 ? "10",                                <ZX>
3600 REPEAT                                 <EN>
3610 # INPUT_LINE                           <EQ>
3620 POSITION 26,8                          <LM>
3630 TRAP #INPUT_LINE:INPUT "";LINE         <BZ>
3640 LINE=ABS(TRUNC(LINE))                  <VS>
3650 UNTIL LINE>=0 AND LINE<32766           <QY>
3660 POSITION 26,8                          <LY>
3670 ? LINE,                                <VI>
3680 REPEAT                                 <FL>
3690 # INPUT_INC                            <GY>
3700 POSITION 26,10                         <PO>
3710 TRAP #INPUT_INC:INPUT "";INC           <KM>
3720 INC=ABS(TRUNC(INC))                    <JT>
3730 UNTIL INC>0                            <WF>
3740 UNTIL LINE+((X+1)*2+1)*INC<32767       <BK>
3750 POSITION 26,10                         <QD>
3760 ? INC,                                 <MZ>
3770 Z=0                                    <AB>
3780 TRAP #ERROR                            <RN>
3790 OPEN #1,8,0,BAS_NAME$                  <YP>
3800 OPEN #2,8,0,INP_NAME$                  <KD>
3810 ? #1;LINE;"▲GRAPHICS▲0"                <QG>
3820 ? #2;LINE;"▲GRAPHICS▲0"                <QS>
3830 LINE=LINE+INC                          <BJ>
3840 ? #2;LINE;"▲OPEN▲#1,4,0,""";EXT_NAM
E$;""""                                     <OC>
3850 LINE=LINE+INC                          <BP>
3860 ? #2;LINE;"▲BGET▲#1,DPEEK(88),959"     <WA>
3870 LINE=LINE+INC                          <BV>
3880 ? #2;LINE;"▲POKE▲93,PEEK(DPEEK(88))
"                                           <WS>
3890 LINE=LINE+INC                          <CB>
3900 I=-1                                   <LN>
3910 X=X+1                                  <TC>
3920 REPEAT                                 <EZ>
3930 Z=Z+1                                  <UK>
3940 REPEAT                                 <FF>
3950 I=I+1                                  <LM>
3960 UNTIL Z(I,0)=Z                         <JF>
3970 DX=Z(I,1) MOD 40                       <BL>
3980 DY=Z(I,1) DIV 40                       <EV>
3990 ? #1;LINE;"▲POSITION▲";DX;",";DY;":
POKE▲99,119"                                <SI>
4000 IF Z(I,2)=0 THEN ? #2;LINE;"▲POSITI
ON▲";DX;",";DY;":POKE▲99,119"               <FD>
4010 LINE=LINE+INC                          <AO>
4020 J=3                                    <WF>
4030 IF Z(I,2)=1                            <UV>
4040 ? #1;LINE;"▲?▲""";                     <YH>
4050 LINE=LINE+INC                          <BA>
4060 REPEAT                                 <EU>
4070 ? #1;CHR$(Z(I,J));                     <NJ>
4080 IF Z(I,J)=34 THEN ? #1;CHR$(34);       <CA>
4090 J=J+1                                  <LV>
4100 UNTIL Z(I,J)=0                         <TG>
4110 ? #1;""";"                             <PA>
4120 ELSE                                   <TF>
4130 A$="DUMMY:"                            <GX>
4140 ? #1;LINE;"▲TRAP▲";LINE-INC;":INPUT
▲"""";";                                    <IH>
4150 ? #2;LINE;"▲TRAP▲";LINE-INC;":INPUT
▲"""";";                                    <IT>
4160 LINE=LINE+INC                          <BF>
4170 REPEAT                                 <EZ>
4180 FLAG=0                                 <UT>
```

```
4190 IF Z(I,J)=40                                    <OV>
4200 IF Z(I,J-1)=36 THEN A$(6,7)="$:"                <II>
4210 ? #1;A$;                                        <QG>
4220 ? #2;A$;                                        <QS>
4230 J=3                                             <WM>
4240 REPEAT                                          <ES>
4250 ? #1;CHR$(Z(I,J));                              <NH>
4260 ? #2;CHR$(Z(I,J));                              <NT>
4270 J=J+1                                           <LT>
4280 UNTIL Z(I,J)=0                                  <UG>
4290 ? #1;"=";A$(1,LEN(A$)-1)                        <HE>
4300 ? #2;"=";A$(1,LEN(A$)-1)                        <GO>
4310 FLAG=1                                          <UN>
4320 ENDIF                                           <SV>
4330 J=J+1                                           <LJ>
4340 UNTIL Z(I,J)=0                                  <TW>
4350 IF FLAG=0                                       <HF>
4360 J=3                                             <WX>
4370 REPEAT                                          <FD>
4380 ? #1;CHR$(Z(I,J));                              <NS>
4390 ? #2;CHR$(Z(I,J));                              <OE>
4400 J=J+1                                           <LC>
4410 UNTIL Z(I,J)=0                                  <TP>
4420 ? #1;CHR$(155);                                 <EU>
4430 ? #2;CHR$(155);                                 <FG>
4440 ENDIF                                           <TD>
4450 ENDIF                                           <TG>
4460 I=-1                                            <LW>
4470 UNTIL Z=X                                       <TV>
4480 TRAP #ERROR                                     <RI>
4490 CLOSE #1                                        <NV>
4500 OPEN #1,8,0,MEM_NAME$                           <DJ>
4510 X=X-1                                           <TN>
4520 PUT #1;X                                        <BC>
4530 FOR I=0 TO X                                    <VI>
4540 FOR J=0 TO 2                                    <WZ>
4550 Z=Z(I,J)                                        <XV>
4560 PUT #1;Z                                        <CO>
4570 IF J=1 THEN PUT #1;Z DIV 256                    <AZ>
4580 NEXT J                                          <GL>
4590 J=J-1                                           <MX>
4600 REPEAT                                          <EO>
4610 J=J+1                                           <LJ>
4620 PUT #1;Z(I,J)                                   <NR>
4630 UNTIL Z(I,J)=0                                  <TZ>
4640 NEXT I                                          <FQ>
4650 CLOSE #1                                        <NN>
4660 DPOKE DL,$0600+120                              <LY>
4670 ?                                               <XF>
4680 ?                                               <XI>
4690 ? "NEW"                                         <JY>
4700 ?                                               <WM>
4710 ?                                               <WP>
4720 ? "ENTER▲""";BAS_NAME$                          <NJ>
4730 ? "ENTER▲""";INP_NAME$                          <AE>
4740 ? "BLOAD▲""";ASM_NAME$                          <JC>
4750 END                                             <QV>
4760 # ERROR                                         <ND>
4770 CLOSE                                           <ZE>
4780 DL_OLD=DPEEK(DL)                                <KM>
4790 DPOKE DL,$0600+140                              <OB>
4800 POKE 764,255                                    <GK>
4810 GET KEY                                         <ON>
4820 DPOKE DL,DL_OLD                                 <KY>
4830 TRAP #ERROR                                     <RB>
4840 GOTO ERL                                        <AB>
4850 DATA▲@@@@@@@editor@@@@@@@                       <MN>
4860 DATA▲nameZ@@@@@@@@@@@@@@@                       <ZH>
4870 DATA▲@@@@@@searching@@@@@                       <ZE>
4880 DATA▲@@@@@@@@order@@@@@@@                       <GV>
4890 DATA▲nameZ@@@@@@@@@@@@@@@                       <ZQ>
4900 DATA▲@@@@@@generate@@@@@@                       <MF>
4910 DATA▲@@@@@@@@ready@@@@@@@                       <BF>
4920 DATA▲@disc@ok_@press@key@                       <QS>
4930 DATA▲@@@@@@@@menu@@@@@@@@                       <YG>
```

Listing 1 (Schluß)

```
Programmname :SDU
Länge        :960 Bytes
--------------------------------

0000:51 52 52 52 52 52 52 52 <5B>
0008:52 52 52 52 52 52 52 52 <B3>
0010:52 52 52 52 52 52 52 52 <CB>
0018:52 52 52 52 52 52 52 52 <F3>
0020:52 52 52 52 52 52 52 45 <EE>
0028:7C 00 00 B3 80 A3 80 B2 <D3>
0030:80 A5 80 A5 80 AE 80 80 <0F>
0038:80 80 A4 80 A5 80 A6 80 <B7>
0040:A9 80 AE 80 A9 80 B4 80 <95>
0048:A9 80 AF 80 AE 00 00 7C <84>
0050:7C 00 00 00 00 00 00 00 <0E>
0058:00 00 00 00 00 00 B5 80 <43>
0060:B4 80 A9 80 AC 80 A9 80 <59>
0068:B4 80 B9 00 00 00 00 00 <99>
0070:00 00 00 00 00 00 00 7C <EC>
0078:41 52 52 52 52 52 52 52 <6B>
0080:52 57 52 52 52 52 52 52 <99>
0088:52 52 52 52 52 52 52 52 <3F>
0090:52 52 52 52 52 52 57 52 <51>
0098:52 52 52 52 52 52 52 44 <60>
00A0:7C 00 00 00 00 00 00 00 <DE>
00A8:00 7C 00 00 00 00 30 72 <59>
00B0:6F 67 72 61 6D 6D 69 65 <B5>
00B8:72 74 00 00 00 00 7C 00 <C7>
00C0:00 00 00 00 00 00 00 7C <3C>
00C8:7C 00 00 00 00 00 00 00 <86>
00D0:00 7C 00 00 00 00 00 00 <EF>
00D8:00 00 00 76 6F 6E 00 00 <61>
00E0:00 00 00 00 00 00 7C 00 <D9>
00E8:00 00 00 00 00 00 00 7C <64>
00F0:7C 00 00 00 00 00 00 00 <AE>
00F8:00 7C 24 61 6E 69 65 6C <B0>
0100:00 33 65 62 61 73 74 69 <BC>
0108:61 6E 65 6C 6C 69 7C 00 <6B>
0110:00 00 00 00 00 00 00 7C <8E>
0118:7C 00 00 00 00 00 00 00 <58>
0120:00 7C 00 00 00 00 00 00 <01>
0128:00 00 10 12 0F 18 17 00 <54>
0130:00 00 00 00 00 00 7C 00 <2B>
0138:00 00 00 00 00 00 00 7C <B6>
0140:7C 00 00 00 00 00 00 00 <00>
0148:00 5A 52 52 52 52 52 52 <0E>
0150:52 52 52 52 52 52 52 52 <09>
0158:52 52 52 52 52 52 43 00 <41>
0160:00 00 00 00 00 00 00 7C <DE>
0168:7C 00 00 00 00 00 00 00 <28>
0170:00 00 00 00 00 00 00 00 <72>
0178:00 00 00 00 00 00 00 00 <7A>
0180:00 00 00 00 00 00 00 00 <82>
0188:00 00 00 00 00 00 00 7C <06>
0190:7C 00 2B 75 72 7A 62 65 <2C>
0198:73 63 68 72 65 69 62 75 <5B>
01A0:6E 67 1A 00 00 00 00 00 <F6>
01A8:00 00 00 00 00 00 00 00 <AA>
01B0:00 00 00 00 00 00 00 7C <2E>
01B8:7C 00 00 00 00 00 00 00 <F8>
01C0:00 00 00 00 00 00 00 00 <C2>
01C8:00 00 00 00 00 00 00 00 <CA>
01D0:00 00 00 00 00 00 00 00 <D2>
01D8:00 00 00 00 00 00 00 7C <56>
01E0:7C 00 00 2D 69 74 00 28 <B4>
01E8:69 6C 66 65 00 64 69 65 <06>
01F0:73 65 73 00 30 72 6F 67 <44>
01F8:72 61 6D 6D 73 00 6B 6F <1F>
0200:65 6E 6E 65 6E 00 00 7C <0D>
0208:7C 00 00 33 69 65 00 6D <C3>
0210:69 74 00 65 69 6E 65 6D <73>
0218:00 26 75 6C 6C 0D 33 63 <7C>
0220:72 65 65 6E 0D 25 64 69 <58>
0228:74 6F 72 00 00 00 00 7C <CC>
0230:7C 00 00 22 21 33 29 23 <DF>
0238:0D 23 6F 64 65 00 66 75 <24>
0240:65 72 00 22 69 6C 64 73 <A6>
0248:63 68 69 72 6D 6D 61 73 <5E>
0250:6B 65 6E 00 00 00 00 7C <2C>
0258:7C 00 00 67 65 6E 65 72 <9C>
0260:69 65 72 65 6E 0E 00 00 <1A>
0268:00 00 00 00 00 00 00 00 <6C>
0270:00 00 00 00 00 00 00 00 <74>
0278:00 00 00 00 00 00 00 7C <F8>
0280:7C 00 00 00 00 00 00 00 <C2>
0288:00 00 00 00 00 00 00 00 <8C>
0290:00 00 00 00 00 00 00 00 <94>
0298:00 00 00 00 00 00 00 00 <9C>
02A0:00 00 00 00 00 00 00 7C <20>
02A8:7C 00 00 00 00 00 00 00 <EA>
02B0:00 00 00 00 00 00 00 00 <B4>
02B8:00 00 00 00 00 00 00 00 <BC>
02C0:00 00 00 00 00 00 00 00 <C4>
02C8:00 00 00 00 00 00 00 7C <48>
02D0:7C 00 00 00 00 26 69 6C <69>
02D8:65 00 2E 61 6D 65 1A 00 <1E>
02E0:00 00 00 00 00 00 00 00 <E4>
02E8:00 2D 65 6D 6F 72 79 1A <E9>
02F0:00 22 23 14 10 00 00 7C <FF>
02F8:7C 00 00 00 00 00 00 00 <DA>
0300:00 00 00 00 00 00 00 00 <06>
0308:00 00 00 00 00 00 00 00 <0E>
0310:00 00 00 00 00 00 00 00 <16>
0318:00 00 00 00 00 00 00 7C <9A>
0320:7C 00 2D 69 6E 69 6D 75 <F7>
0328:6D 00 23 6F 64 65 1A 00 <19>
0330:2E 00 00 00 00 00 00 00 <4D>
0338:00 00 00 2D 61 72 6B 1A <C4>
0340:00 20 00 00 00 00 00 7C <CA>
0348:7C 00 00 00 00 00 00 00 <0C>
0350:00 00 00 00 00 00 00 00 <56>
0358:00 00 00 00 00 00 00 00 <5E>
0360:00 00 00 00 00 00 00 00 <66>
0368:00 00 00 00 00 00 00 7C <EA>
0370:5A 52 52 52 52 52 52 52 <D1>
0378:52 52 52 52 52 52 52 52 <DD>
0380:52 52 52 52 52 52 52 52 <35>
0388:52 52 52 52 52 52 36 65 <28>
0390:72 73 0E 12 0E 10 52 43 <23>
0398:00 00 00 00 00 00 00 00 <9E>
03A0:00 00 00 00 00 00 00 00 <A6>
03A8:00 00 00 00 00 00 00 00 <AE>
03B0:00 00 00 00 00 00 00 00 <B6>
03B8:00 00 00 00 00 00 00 00 <BE>
```

Listing 2. »SDU«. Bitte mit AMPEL eingeben.

Prüfsummer paßt auf

Unser Prüfsummer für alle Atari-Computer ist besonders komfortabel. Mit Hilfe dieses Programms lassen sich Listings auf Anhieb korrekt und problemlos eingeben.

Mag der Atari-Prüfsummer auf den ersten Blick etwas umfangreich erscheinen, der Aufwand lohnt sich. So legt der Prüfsummer zum Beispiel eine zusätzliche Statuszeile am oberen Bildschirmrand an. Diese Zeile ist von Basic aus nicht zu erreichen, also auch nicht zu löschen. Weiterhin können Sie den Prüfsummer auch für die Eingabe von Turbo-Basic-XL-Programmen verwenden. Beachten Sie hierzu die besonderen Hinweise!

In dem abgedruckten Listing finden Sie gewisse Zeichen vor, die der Atari nicht kennt. Es handelt sich dabei einerseits um Dreiecke und andererseits um geschweifte Klammern. Die Dreiecke stellen grundsätzlich Leerstellen dar; Texte, die zwischen geschweiften Klammern stehen, repräsentieren Atari-spezifische Grafik- und Sonderzeichen. Ein Beispiel: »ESC CTL=« entspricht dem ASCII-Zeichen 29. Um dieses Zeichen einzugeben, betätigen Sie zunächst die <ESC>-Taste und dann gleichzeitig die <CONTROL>- und die <=>-Taste. Daraufhin erscheint das gewünschte Zeichen auf dem Bildschirm. Eine komplette Aufstellung aller Grafik- und Sonderzeichen finden Sie in der Tabelle. Weiterhin gibt es unterstrichene Zeichen, die eine inverse Darstellung bedeuten. Bevor Sie ein solches Zeichen eingeben, müssen Sie unbedingt die Atari-Taste betätigen.

Bild 1 zeigt eine Programmzeile in der üblichen Darstellungsweise und zum Vergleich dazu eine in konvertierter Form.

```
4850 POSITION 11,2:? "└──────────────┘":
POSITION 4,5:? "FILENAME      SIZE":POSITI
ON 2,7

4850 POSITION 11,2:? "{CTL Z}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{SHIFT =}":POSITION 4,5:? "FILENA
ME▲▲▲▲SIZE":POSITION 2,7
```

Bild 1. Oben eine Programmzeile, wie sie auf dem Bildschirm erscheint. Darunter dieselbe Zeile in der konvertierten Form.

Geben Sie also zunächst das Prüfsummenprogramm von Basic - oder Turbo-Basic XL - aus ein. Zwar werden sämtliche DATA-Werte nach dem Start mit RUN überprüft, aber »nobody is perfect«.

Kontrollieren Sie also das Prüfsummenprogramm vor dem Start noch einmal. Speichern Sie es dann auf einem Datenträger. Jetzt können Sie das Programm beruhigt mit RUN starten. Aus den DATA-Werten wird nun ein Maschinen-Programm erzeugt. Sollte sich beim Eintippen des Programms ein falscher DATA-Wert eingeschlichen haben, so wird die entsprechende Zeilennummer auf dem Bildschirm angezeigt.

Nach der Korrektur starten Sie das Programm erneut mit RUN. Wurde das Programm fehlerlos eingegeben, stehen folgende Funktionen zur Auswahl:

1. Das Prüfsummen-Programm als Maschinensprache-File auf Diskette schreiben. Nach Betätigung der Taste 1 fragt das Programm nach dem Namen für das Prüfsummenprogramm. Tippen Sie also beispielsweise »PRUEFER.OBJ« ein, so wird nach Druck auf die <RETURN>-Taste das Prüfsummenprogramm auf Diskette gespeichert. Der Computer schließt den Vorgang mit der »READY«-Meldung ab.

Um das Prüfcode-Programm verwenden zu können, löschen Sie den Programmspeicher zunächst mit NEW. Vom DOS-Menü aus (einfach DOS eintippen und die <RETURN>-Taste betätigen) wählen Sie die Option L und geben den Namen des vorher erzeugten Maschinenprogramms ein. Jetzt wird das Prüfsummenprogramm in den RAM-Speicher geladen. Mit <B> gelangen Sie wieder zurück ins Basic.

```
Pruefcode :  >  <  (c) 85 by W. Kress/wb
 1000 PROC RECHNEN
 1010    GRAPHICS 15+16:POKE 764,255
 1030    DX=(XMAX-XMIN)/159
 1040    DY=(YMAX-YMIN)/191
 1050    CX=XMIN:CY=YMAX
 1080    FOR ZEILE=0 TO 191
 1090      FOR SPALTE=0 TO 159
 1110        TI=0:XW=0:YW=0:X2=0:Y2=0
 1120        WHILE TI<TMAX AND (X2+Y2)<8

 1130          YW=2*XW*YW-CY:XW=X2-Y2-CX

 1150          X2=XW^2:Y2=YW^2:TI=TI+1
 1180        WEND
 1190        IF TI=TMAX
 1200          COLOR %0
 1220        ELSE
 1230          COLOR (TI MOD 3)+%1
 1240        ENDIF
 1250        CX=CX+DX:PLOT SPALTE,ZEILE
 1260        IF PEEK(764)=28 THEN POP :G
 OTO 1295
 1270      NEXT SPALTE
 1280      C
```

Bild 2. So sollte der Bildschirm nach dem Aktivieren des Prüfsummers aussehen

Aktiviert wird jetzt das Prüfsummenprogramm mit »?USR(1536)«. Am oberen Bildschirmrand erhalten Sie daraufhin eine zusätzliche Zeile mit dem Schriftzug »Prüfcode: > < (c) by W. Kress/wb«. Wie die Aufteilung des Bildschirms unter Verwendung des Prüfsummenprogramms aussieht, zeigt Bild 2.

2. Das Maschinenprogramm wird als Boot-File auf Kassette geschrieben. Das spätere Laden oder Booten geht folgendermaßen vor sich:
- bei ausgeschaltetem Computer die <START>-Taste drücken und Computer einschalten,
- nach Ertönen des Summtons die <START>-Taste loslassen,
- nacheinander die <PLAY>-Taste des Kassettenrecorders und die <RETURN>-Taste betätigen,
- das Prüfsummenprogramm wird geladen und automatisch gestartet

3. Programm beenden. Das Maschinenprogramm wird nicht gespeichert.

Das Prüfsummenprogramm wird in der Page 6 abgelegt, einem vor NEW geschützten Speicherbereich. Abschalten kann man es durch Drücken von <SYSTEM RESET>, ein Neustart erfolgt mit dem Aufruf: »?USR(1536)«.

Sobald das Prüfsummenprogramm aktiviert ist, können beim Zugriff auf ein Speichermedium Bildstörungen auftreten. Um dies zu vermeiden, brauchen Sie nach dem Laden oder nach dem Aufruf mit »?USR(1536)« nur »POKE 39998,30« einzugeben.

Wurde das Programm von Kassette gebootet, wird es jedesmal nach Betätigung der <SYSTEM RESET>-Taste

erneut gestartet. Gibt man jedoch vor der Betätigung der <SYSTEM RESET>-Taste »POKE 9,0« ein, kann man diesen Effekt umgehen.

Bevor Sie nun ein eingetipptes Basic- oder Turbo-Basic XL-Programm mit RUN starten, sollten Sie es sicherheitshalber noch einmal speichern. Weiterhin sollte das Prüfsummen-Programm mit <SYSTEM RESET> desaktiviert werden, da auch manche Basic-Programme die Page 6 für Maschinen-Unterprogramme verwenden. Es könnte sonst zu Überschneidungen und zum Systemabsturz kommen.

Der Prüfcode besteht jeweils aus zwei Zeichen. Er setzt sich aus den Buchstaben von A bis Z zusammen. Geben Sie nun probeweise eine Programmzeile ein und schließen Sie mit der <RETURN>-Taste ab. Der zugehörige Prüfcode erscheint dann in der obersten Zeile. Sollte der Code nicht mit dem im abgedruckten Listing übereinstimmen, kann die Programmzeile sofort verbessert werden. Die jeweilige Programmzeile braucht dazu nicht erneut eingegeben zu werden, da sämtliche Editierfunktionen des Atari-Computers zur Verfügung stehen.

Ein wichtiger Hinweis:
Für die erste Programmzeile kann eventuell ein falscher Prüfcode errechnet werden. Sollte die Prüfsumme also nicht mit der im Listing abgedruckten übereinstimmen, fahren Sie mit dem Cursor nochmals auf die erste Zeile. Sobald Sie die <RETURN>-Taste betätigt haben, erscheint der richtige Code auf dem Bildschirm. Stimmt der ausgegebene Wert dann immer noch nicht, überprüfen Sie nochmals jedes einzelne Zeichen der von Ihnen eingegebenen Programmzeile.

Bei der Eingabe von Programmen sind die üblichen Abkürzungen für Basic-Befehle zulässig. Sie können also zum Beispiel G. für GOTO, SE. für SETCOLOR und so weiter verwenden. Eine Ausnahme gilt jedoch bei der Abkürzung für den PRINT-Befehl, den man in der Regel mit einem Fragezeichen abkürzen kann. **Verwenden Sie das Fragezeichen bitte nur dann, wenn es auch im Listing verwendet wird.**

Möchten Sie nachträglich ein Programm überprüfen, so laden Sie zuerst das Prüfsummenprogramm und starten es mit »?USR(1536)«. Anschließend laden Sie Ihr Programm und listen einen Teil auf dem Bildschirm. Dann bewegen Sie den Cursor an irgendeine Stelle einer Programmzeile. Mit Betätigung der <RETURN>-Taste wird der Prüfcode in der Statuszeile dargestellt. Stimmt der Prüfcode mit dem im beigefügten Listing überein, betätigen Sie wieder die <RETURN>-Taste und der Prüfcode der nächsten Zeile wird errechnet und angezeigt. Auf diese Art und Weise können Sie ein komplettes Programm oder auch nur einzelne Zeilen »durchchecken«.

Sollte der Code nicht mit dem des Listings übereinstimmen, überprüfen Sie Ihre Programmzeile. Denn bei allen abgetippten Listings gilt: Die Programme müssen so abge-

Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)
A	B	C	A	B	C	A	B	C	A	B	C
0		CTL ,	19		CTL S	128		(Invers) CTL ,	147		(Invers) CTL S
1		CTL A	20		CTL T	129		(Invers) CTL A	148		(Invers) CTL T
2		CTL B	21		CTL U	130		(Invers) CTL B	149		(Invers) CTL U
3		CTL C	22		CTL V	131		(Invers) CTL C	150		(Invers) CTL V
4		CTL D	23		CTL W	132		(Invers) CTL D	151		(Invers) CTL W
5		CTL E	24		CTL X	133		(Invers) CTL E	152		(Invers) CTL X
6		CTL F	25		CTL Y	134		(Invers) CTL F	153		(Invers) CTL Y
7		CTL G	26		CTL Z	135		(Invers) CTL G	154		(Invers) CTL Z
8		CTL H	27		ESC ESC	136		(Invers) CTL H	156		ESC SHIFT DEL
9		CTL I	28		ESC CTL –	137		(Invers) CTL I	157		ESC SHIFT >
10		CTL J	29		ESC CTL =	138		(Invers) CTL J	158		ESC CTL TAB
11		CTL K	30		ESC CTL +	139		(Invers) CTL K	159		ESC SHIFT TAB
12		CTL L	31		ESC CTL *	140		(Invers) CTL L	224		(Invers) CTL .
13		CTL M	96		CTL .	141		(Invers) CTL M	251		(Invers) CTL ;
14		CTL N	123		CTL ;	142		(Invers) CTL N	252		(Invers) SHIFT =
15		CTL O	124		SHIFT =	143		(Invers) CTL C	253		ESC CTL 2
16		CTL P	125		ESC CTL <	144		(Invers) CTL P	254		ESC CTL DEL
17		CTL Q	126		ESC DEL	145		(Invers) CTL Q	255		ESC CTL >
18		CTL R	127		ESC TAB	146		(Invers) CTL R			

Tabelle der Atari-Grafikzeichen

tippt werden, wie sie gedruckt sind. Stimmt die Anzahl der Leerzeichen? Haben Sie eventuell Zeichen vertauscht? Oder haben Sie ein »?« anstelle eines PRINT-Befehls eingegeben? Haben Sie den Fehler gefunden? Dann weiterhin viel Erfolg mit Ihren Atari-Programmen!

Besondere Hinweise zum Prüfsummenprogramm

Wenn Sie ein Turbo-Basic XL-Programm eingeben möchten, müssen Sie das Prüfsummenprogramm mit einem Hilfsprogramm starten. Weiterhin empfiehlt es sich, eine separate Diskette sozusagen als Eingabediskette zu verwenden. Formatieren Sie also eine Diskette. Verwenden Sie aber bitte **nur DOS 2.0 oder 2.5!** Speichern Sie anschließend die DOS-Files und Turbo-Basic XL mit dem Namen »AUTORUN.SYS« auf dieser Diskette. Dann schalten Sie den Computer aus und booten die soeben angelegte Diskette. Geben Sie anschließend folgende Programmzeile ein:

```
1 BLOAD "D:PRUEFER.OBJ":POKE $BC3E,$1E:*L-:NEW
```

Speichern Sie diese Programmzeile dann mit »SAVE "D:AUTORUN.BAS"« auf Ihrer Eingabediskette. Dieses kurze Programm sorgt beim Booten des Computers dafür, daß das Prüfsummenprogramm automatisch geladen und ausgeführt wird. Achtung: **Das Prüfsummenprogramm läßt sich nicht vom DOS-Menü aus starten! Die oben aufgeführte Programmzeile muß unbedingt beim Bootvorgang ausgeführt werden.** Auf Ihrer Prüfsummer-Diskette sollten sich also folgende Files befinden:

1. DOS.SYS (entweder DOS 2.0 oder 2.5)
2. DUP.SYS (passend zum DOS.SYS-File)
3. AUTORUN.SYS (Turbo-Basic XL)
4. AUTORUN.BAS (das Initialisierungsprogramm)
5. PRUEFER.OBJ (das eigentliche Prüfsummenprogramm als Maschinencode-File)

Wenn Sie Turbo-Basic unter unserem Happy-DOS verwenden, müssen die einzelnen Files anders auf die Diskette geschrieben werden. Die folgende Liste zeigt wie:

1. Diskette formatieren mit »FO#«
2. DOS.SYS auf die Diskette schreiben mit »IN#«
3. Turbo-Basic mit »COPY AUTORUN.SYS« auf die initialisierte Diskette schreiben
4. Turbo-Basic mit »REN AUTORUN.SYS TURBO.COM« auf der Diskette umbenennen
5. Mit »JOB TURBO« das Happy-DOS veranlassen, das Turbo-Basic automatisch zu laden
6. »AUTORUN.BAS« wie in der ersten Liste beschrieben auf die Diskette schreiben
7. »PRUEFER.OBJ« auf die Diskette schreiben

Das Turbo-Basic wird dann automatisch geladen, gestartet und der Prüfsummer installiert.

Eine ähnlich Prozedur muß angewendet werden, wenn andere DOS-Versionen, wie zum Beispiel OSS CP/A 1.2, 2.1 oder DOS XL, benutzt werden. Anstelle von Punkt 5 müssen Sie allerdings von Basic aus die Befehlsfolge »OPEN #1,8,0,"D:STARTUP:EXC":? #1; "TURBO": END« eingeben. Dann wird auch in diesen DOS-Versionen Ihre Prüfsummer-Diskette automatisch gestartet.

Basic-Programme schnell und zuverlässig eingegeben

Um also unter Turbo-Basic XL den Prüfsummer verwenden zu können, brauchen Sie nur Ihre Prüfsummer-Diskette ins Laufwerk einzuschieben und Ihren Computer einzuschalten. Alles weitere erledigt der Computer von selbst. Vergessen Sie bitte nicht, sich von der neuen Prüfsummer-Version auch eine Sicherheitskopie anzulegen. Sind noch Fragen offen, wenden Sie sich bitte an die Redaktion.

(Werner Breuer/hf)

Steckbrief	
Programm:	Prüfsummer
Sprache:	Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Kassette, Diskette

```
100 REM ******************************   <LU>
110 REM *                            *   <HA>
120 REM *   PRUEFCODE fuer ATARI     *   <CU>
130 REM *                            *   <HE>
140 REM *  von  W.Kress              *   <ZZ>
150 REM *       Lindenweg 17         *   <XA>
160 REM *       7590 Achern          *   <IO>
170 REM *  (c)  by Happy-Computer    *   <DC>
180 REM *                            *   <HO>
190 REM ******************************   <MM>
200 REM                                  <AJ>
210 DIM F$(15),SED$(2),D$(504),C$(11)    <RC>
220 C$="{CTL ,}{CTL D}u7{CTL A}{CTL F})<
{CTL M}{CTL B}S"                         <IZ>
230 GRAPHICS 0:POKE 82,0:POKE 752,1      <KV>
240 POSITION 12,10:POKE 201,8            <BO>
250 ? "Bitte▲warten▲!":POKE 752,0        <DX>
260 Z=1000                               <GZ>
270 FOR I=1 TO 497 STEP 8                <OL>
280 PSUMM=0                              <MG>
290 FOR J=0 TO 7                         <PK>
300 READ SED$                            <ZV>
310 H=ASC(SED$(1))-48                    <EQ>
320 IF H>9 THEN H=H-7                    <LT>
330 L=ASC(SED$(2))-48                    <GE>
340 IF L>9 THEN L=L-7                    <SR>
350 DEZ=H*16+L:D$(I+J)=CHR$(DEZ)         <JV>
360 PSUMM=PSUMM+DEZ                      <MV>
370 NEXT J                               <GR>
380 READ ZSUMM                           <AW>
390 IF PSUMM=ZSUMM THEN 410              <IN>
400 ? "{ESC CTL 2}DATENFEHLER▲!":LIST Z:
END                                      <YX>
410 GSUMM=GSUMM+PSUMM:Z=Z+10             <NJ>
420 NEXT I                               <FY>
430 READ ESUMM                           <SL>
440 IF ESUMM=GSUMM THEN 460              <ZO>
450 ? "{ESC CTL 2}FEHLENDE▲ODER▲DOPPELTE
▲DATA-ZEILE":END                         <IZ>
460 ? "{ESC CTL 2}"                      <EX>
470 ? "{ESC CTL <}"                      <RP>
480 POSITION 2,6:? :POKE 752,1           <UQ>
490 ? ,"{CTL Y}1{CTL Y}▲Auf▲DISKETTE▲sch
reiben{ESC CTL =}{ESC CTL =}"            <AY>
500 ? ,"{CTL Y}2{CTL Y}▲Auf▲CASSETTE▲sch
reiben{ESC CTL =}{ESC CTL =}"            <AQ>
510 ? ,"{CTL Y}3{CTL Y}▲ENDE"            <VR>
520 OPEN #1,4,0,"K":GET #1,K:CLOSE #1    <NQ>
530 K=K-48:IF K<1 OR K>3 THEN ? "{ESC CT
L 2}";:GOTO 520                          <WU>
540 POSITION 3,4+3*K:? "--->";           <FU>
550 POSITION 2,20:? "{ESC SHIFT DEL}";   <EW>
560 ON K GOTO 570,590,680                <DI>
570 POKE 752,0:? "{ESC TAB}DATEINAME▲:▲▲
```

Listing zum Atari-Prüfsummer



```
D1:{ESC CTL +}{ESC CTL +}{ESC CTL +}{ESC
 CTL +}";                                   <KM>
580 POKE 764,44:INPUT F$:D=1:GOTO 610       <ZI>
590 F$="C:":D=0                             <MT>
600 ? ,"CASSETTE.FERTIG.MACHEN,.RETURN";    <AH>
610 TRAP 660                                <OJ>
620 OPEN #1,8,128,F$                        <YW>
630 IF D THEN ? #1;D$(1,500);               <UB>
640 IF  NOT D THEN ? #1;C$;D$(7,494)        <XK>
650 CLOSE #1:GOTO 470                       <EW>
660 ? "{ESC CTL 2}{ESC CTL -}{ESC SHIFT
DEL}{ESC TAB}I/O.-.FEHLER.";PEEK(195);      <CR>
670 TRAP 40000:CLOSE #1:GOTO 480            <HQ>
680 POKE 752,0:END                          <YW>
690 REM ----------------------------        <ZP>
1000 DATA.FF,FF,00,38,E7,39,A2,FF,1271      <EA>
1010 DATA.BD,EB,38,9D,00,06,CA,E0,1069      <NF>
1020 DATA.FF,D0,F5,A2,60,BD,8A,38,1349      <VW>
1030 DATA.9D,00,01,CA,10,F7,A2,0F,800       <OL>
1040 DATA.BD,00,E4,9D,61,01,CA,10,890       <LI>
1050 DATA.F7,18,AD,65,01,69,01,8D,793       <MP>
1060 DATA.6B,06,AD,66,01,69,00,8D,635       <CE>
1070 DATA.6C,06,A9,2A,8D,65,01,A9,737       <XW>
1080 DATA.06,8D,66,01,AD,67,01,8D,668       <VG>
1090 DATA.30,06,AD,68,01,8D,34,06,531       <HH>
1100 DATA.38,AD,30,02,E9,02,8D,02,657       <ZR>
1110 DATA.06,8D,11,06,AD,31,02,E9,627       <EV>
1120 DATA.00,8D,07,06,8D,12,06,A9,488       <DM>
1130 DATA.45,20,7A,38,90,11,8E,26,620       <CP>
1140 DATA.06,A9,00,20,7A,38,90,07,536       <YQ>
1150 DATA.8E,17,06,20,01,06,18,60,330       <QZ>
1160 DATA.A2,00,DD,1A,03,F0,08,E8,892       <CH>
1170 DATA.E8,E8,E0,21,D0,F4,18,60,1293      <CK>
1180 DATA.70,42,39,01,9E,06,9E,06,564       <UJ>
1190 DATA.9E,06,75,06,9E,06,9E,06,615       <EX>
1200 DATA.00,00,00,00,4C,2E,22,54,240       <CY>
1210 DATA.3A,22,2C,50,45,45,4B,28,469       <TN>
1220 DATA.31,31,33,29,2B,32,35,36,390       <LL>
1230 DATA.2A,50,45,45,4B,28,31,31,473       <SG>
1240 DATA.34,29,9B,01,61,45,01,04,420       <SW>
1250 DATA.54,30,72,75,65,66,63,6F,776       <YW>
1260 DATA.64,65,00,1A,00,00,1E,00,257       <ER>
1270 DATA.00,1C,00,00,08,63,09,00,144       <LV>
1280 DATA.18,15,00,62,79,00,37,0E,333       <NU>
1290 DATA.00,2B,72,65,73,73,0F,77,622       <KV>
1300 DATA.62,68,A9,FF,8D,30,02,A9,986       <OK>
1310 DATA.FF,8D,31,02,A0,03,B9,00,795       <ZL>
1320 DATA.01,99,FF,FF,88,10,F7,A0,1223      <DR>
1330 DATA.FF,A2,05,BD,33,01,99,1A,842       <OO>
1340 DATA.03,C8,E0,03,D0,02,A0,FF,1055      <QQ>
1350 DATA.CA,10,F0,60,A5,77,F0,0A,1088      <OT>
1360 DATA.A9,FF,85,26,A9,FF,85,27,1191      <LE>
1370 DATA.C6,77,A5,73,F0,2D,A5,A1,1208      <VB>
1380 DATA.10,04,C6,73,F0,25,A6,78,896       <NG>
1390 DATA.BD,14,01,E6,78,C9,9B,D0,1124      <ZM>
1400 DATA.4F,A2,00,86,78,86,73,A6,910       <WM>
1410 DATA.A0,86,71,A6,A1,86,72,A2,1144      <QR>
1420 DATA.9E,86,26,A2,06,86,27,E6,901       <AY>
1430 DATA.77,D0,35,20,3E,F6,08,C9,929       <CM>
1440 DATA.9B,D0,02,E6,73,28,60,A2,1008      <IO>
1450 DATA.00,C9,9B,F0,26,86,CC,A6,1138      <LN>
1460 DATA.00,86,D1,85,CD,A0,08,46,919       <OZ>
1470 DATA.D1,90,0C,18,A2,03,B5,CD,940       <DL>
1480 DATA.75,CA,95,CD,CA,D0,F7,06,1336      <ZZ>
1490 DATA.CD,26,CC,88,D0,E9,E6,00,1254      <EW>
1500 DATA.A0,01,60,A9,A4,A0,02,20,784       <HA>
1510 DATA.CC,06,A2,03,B5,CA,95,CD,1112      <MP>
1520 DATA.CA,D0,F9,A9,1A,20,CC,06,1096      <IN>
1530 DATA.A5,CD,18,69,21,8D,48,01,746       <BX>
1540 DATA.A5,D0,69,21,8D,47,01,84,856       <UQ>
1550 DATA.D0,C8,84,00,60,85,72,84,1015      <LY>
1560 DATA.71,86,CB,86,CC,86,CD,A0,1287      <JS>
1570 DATA.18,18,A2,05,36,CB,CA,10,690       <RJ>
1580 DATA.FB,38,A2,03,B5,CA,F5,6F,1211      <RY>
1590 DATA.95,73,CA,D0,F7,90,0B,B5,1257      <GI>
1600 DATA.74,95,CB,E8,E0,03,D0,F7,1382      <II>
1610 DATA.E6,D0,88,D0,DC,60,E2,02,1326      <JV>
1620 DATA.E3,02,00,38,00,00,00,00,285       <BR>
1630 DATA.53423                             <NQ>
```

Listing zum Atari-Prüfsummer (Schluß)

Ampel-Version 1.1

Sie möchten unser neuestes Listing abtippen? Mit AMPEL wird das Eingeben von reinen Atari-Maschinenprogrammen zum Kinderspiel.

Speziell bei Maschinenprogrammen kommt es darauf an, daß jedes Byte korrekt eingegeben wird. Schon ein einziger falscher Wert führt meist dazu, daß das eingegebene Programm nicht läuft. Dies hat dann oft eine stundenlange Fehlersuche zur Folge. Deshalb haben wir für Sie »AMPEL« (Atari-Maschinen-Programm-Eingabe-Listing) entwickeln lassen. Und damit das Eintippen für Sie möglichst einfach und reibungslos verläuft, wählen wir eine spezielle Darstellungsform für Maschinenprogramme. Beachten Sie bitte, daß Sie zukünftig unbedingt AMPEL-Version 1.1 verwenden müssen, um Atari-Maschinenprogramme einzugeben. Von Basic aus lassen sich solche Listings nicht eingeben. Verwahren Sie also das Programm zu AMPEL sorgfältig und legen Sie eine zusätzliche Kopie an einen sicheren Platz.

Geben Sie zunächst das nachfolgende Listing ein. Verwenden Sie dazu den Atari-Prüfsummer. Bevor Sie AMPEL mit RUN starten, sollten Sie das eingegebene Basic-Programm auf Diskette oder Kassette speichern. Hat sich nämlich ein Fehler im Maschinensprache-Teil eingeschlichen, kann Ihr Computer abstürzen. In dem Fall müßte AMPEL erneut eingegeben werden.

Nachdem Sie das Programm mit RUN gestartet haben, werden zuerst die DATA-Werke eingePOKEt. Dieser Vorgang nimmt einige Sekunden in Anspruch. Danach müssen Sie die Länge des einzugebenden Maschinenprogramms eingeben. Diesen Wert entnehmen Sie bitte am Ende des abgedruckten Maschinenprogramms. Anschließend geben Sie den Programmnamen an, unter dem Ihr Programm gespeichert wird. Auf dem Bildschirm erscheint jetzt »0000:«, die Aufforderung, mit der Eingabe der ersten Zeile des AMPEL-Listings zu beginnen.

Betrachten wir eine AMPEL-Listing-Zeile aus der Nähe:

```
0000:00 02 00 07 06 07 20 BD <F3>
```

Die ersten vier Zahlen (hier 0000) stellen sozusagen die Zeilennummer dar. Diese Zahlen sowie den darauffolgenden Doppelpunkt brauchen Sie nicht einzugeben. Danach folgen acht zweistellige Hexadezimalzahlen, die Sie dem Listing entnehmen. Die Eingabe erfolgt ohne Betätigung der <RETURN>-Taste. Der Cursor springt selbständig von einer Position zur nächsten. Leerzeichen sowie die beiden Zeichen »<« und »>« fügt AMPEL automatisch ein. Ist die Prüfsumme korrekt – sie muß auch von Ihnen eingegeben werden – springt der Cursor in die nächste Zeile. Falls nicht, ertönt ein akustisches Signal und alle acht Werte, inklusive der Prüfsumme, müssen nochmals eingegeben werden. Dadurch wir gewährleistet, daß wirklich alle eingegebenen Hexadezimalzahlen im Listing stimmen und sich keine unbemerkten Fehler einschleichen können.

Außer den Hex-Tasten (<0> bis <9> und <A> bis <F>), wird nur noch <DELETE BACKSPACE> zum Korrigieren des letzten Zeichens und <CONTROL S> abgefragt.

Zwischenspeichern

Da Sie ein umfangreiches Maschinenprogramm sicher nicht an einem Tag eintippen möchten, können Sie zu jedem Zeitpunkt mit <CONTROL S> zwischenspeichern. Dazu betätigen Sie einfach <CONTROL S> und der Code wird unter dem zuvor eingegebenen Programmnamen gespeichert. Weiterhin wird automatisch die jeweils zuletzt gespeicherte Version in »BACKUP.OBJ« umbenannt. Zuvor wird natürlich eine ältere »BACKUP.OBJ«-Datei gelöscht. Danach können Sie den Computer ausschalten oder AMPEL mit <RESET> verlassen.

Wenn Sie das nächstemal AMPEL starten, muß wieder die Programmlänge und der entsprechende Name, mit dem Sie Ihre letzte Version gespeichert haben, eingegeben werden. Existiert bereits eine Datei mit dem entsprechenden Namen, wird die Datei automatisch geladen. Weiterhin wird noch die Zeilennummer auf dem Bildschirm angezeigt, ab der Sie mit der Eingabe der Hex-Werte fortfahren müssen.

Speichern Sie lieber öfter mit <CONTROL+S>, um auch gegen eventuelle Stromausfälle gewappnet zu sein. Weiterhin empfiehlt es sich aus Sicherheitsgründen, zwei Disketten für Daten vorzusehen. Wenn Sie das komplette Programm eingegeben haben, speichert der Computer nach Eingabe des letzten Bytes das Programm auf Diskette und meldet sich danach mit der READY-Meldung.

Wenn Sie Ihr Maschinenprogramm anschließend laden möchten, erfolgt dies vom DOS-Menü aus. Wählen Sie hier die L-Funktion und geben Sie anschließend den entsprechenden Programmnamen ein.

Besondere Hinweise für Kassettenrecorder-Besitzer

Damit AMPEL auch einwandfrei mit einem Kassettenrecorder funktioniert, müssen folgende Programmzeilen geändert werden:

```
290 F$="C:"   :AD=0
300 ? " Altes File laden J/N "  ;
310 INPUT FR$
320 IF FR$<>"J" THEN 510
420 TRAP 510:OPEN #1,4,128,F$
590 TRAP 660:CLOSE#1:OPEN #1,8,128,F$
```

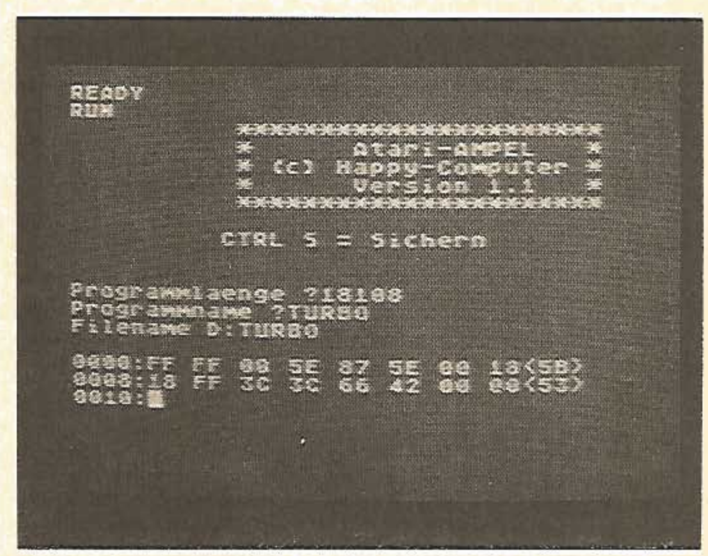
Die Zeilen 330 bis 360 entfallen, ebenso die Zeilen 450, 600 und 610.

Speichern Sie anschließend die geänderte AMPEL-Version mit CSAVE. Somit haben Sie das benötigte Eingabeprogramm bereits vorliegen.

Sicherheit über alles

Mit <CONTROL S> kann zu jedem Zeitpunkt gespeichert werden. Daraufhin erklingt ein doppelter Signalton. Betätigen Sie dann noch die <RECORD>- und <PLAY>-Tasten des Recorders und anschließend die <RETURN>-Taste. Der Speichervorgang beginnt. Wenn Sie dann mit einem anderen Programm arbeiten möchten, gelangen Sie mit <RESET> zurück ins Basic.

Wenn Sie zu einem späteren Zeitpunkt mit der Arbeit fortfahren möchten, laden Sie wieder zuerst AMPEL, starten es mit RUN, geben die Programmlänge ein und beantworten die Frage »Altes File laden J/N« mit J. Daraufhin meldet sich der Computer wieder mit einem Brummton. Spulen Sie das Band an den Anfang zurück, betätigen Sie dann noch die <PLAY>-Taste und <RETURN>. Jetzt wird der zuletzt eingegebene Programmteil geladen. Anschließend meldet sich der Atari-Computer wieder mit der Zeilennummer, ab der Sie mit der Eingabe fortfahren müssen.



So sieht der Bildschirm bei der Eingabe von Maschinenprogrammen aus. Zwischenspeichern kann man jederzeit mit der Tastenkombination <CONTROL+S>.

Um später das eingegebene Programm zu starten, muß stets zuerst das »Kassetten-Hilfsprogramm« geladen werden. Nach dem Ladevorgang meldet sich der Computer mit dem Brummton. Anschließend legen Sie bitte Ihre Kassette mit Ihrem Maschinenprogramm in den Recorder ein, drücken die <PLAY>-Taste des Recorders und während Sie die <START>- und <OPTION>-Tasten des Computers drücken, schalten Sie die Stromversorgung ein. Nach dem Brummton nur noch <RETURN> drücken und Ihr Programm wird geladen und anschließend automatisch gestartet.

Programme schnell geladen

Speziell bei Kassettenrecordern kann es Probleme beim Laden von Maschinenprogrammen geben. Deshalb sollten Sie stets mehrere Sicherheitskopien von Ihren Programmen machen. Hinzu kommt, daß das Laden von Kassette sehr viel Zeit beanspruchen kann, da die Übertragungsgeschwindigkeit sehr langsam ist. Besitzer einer Diskettenstation haben es hier viel besser. Sollten Sie mit AMPEL auf Schwierigkeiten stoßen, wenden Sie sich an die Redaktion.

(Frank Ostrowski/Werner Breuer/hf)

Steckbrief	
Programm:	AMPEL
Sprache:	Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Kassette, Diskette



```
100 ? ,"▲**********************"               <HI>
110 ? ,"▲*▲▲▲▲▲▲Atari-AMPEL▲▲▲*"               <XO>
115 ? ,"▲*▲(c)▲Happy-Computer▲*"               <JT>
117 ? ,"▲*▲▲▲▲▲▲Version▲1.1▲▲▲*"               <OE>
120 ? ,"▲**********************"               <HM>
130 ?                                          <JU>
140 ? ,"CTRL▲S▲=▲Sichern"                      <ZY>
150 ?                                          <JY>
160 ? "Einen▲Moment▲Geduld▲bitte.";            <CL>
190 POKE 16,64:POKE 53774,64                   <OQ>
230 DIM F$(15),FB$(15),FR$(27)                 <WK>
240 DIM ML$(261),CIO$(83)                      <MH>
250 GOSUB 1000                                 <QT>
260 ? CHR$(156)                                <DG>
270 ? "Programmlaenge▲";:INPUT L               <EH>
280 DIM A$(L+8)                                <GW>
290 ? "Programmname▲";:INPUT F$                <IR>
300 IF F$="" THEN 290                          <MA>
310 F=0:FOR I=1 TO LEN(F$):F=F+(F$(I,I)=
":"):NEXT I                                    <IR>
320 IF F=0 THEN FR$="D:":FR$(3)=F$:F$=FR
$                                              <UE>
330 ? "Filename▲";F$                           <TA>
340 FB$="D:BACKUP.OBJ"                         <NW>
350 FR$=F$:FR$(LEN(FR$)+1)=","                 <XT>
360 FR$(LEN(FR$)+1)=FB$(3)                     <IO>
420 AD=0:TRAP 510:OPEN #1,4,0,F$               <QM>
430 E=USR(ADR(CIO$),1,7,ADR(A$),L)             <ZH>
440 IF E=1 THEN ? "Schon▲fertig":END           <IN>
450 IF E<>136 THEN ? "Disk-Read-Error▲";
E:END                                          <QL>
460 AD=PEEK(856)+256*PEEK(857)                 <QT>
470 AD=8*INT(AD/8):TRAP 65535                  <VQ>
510 CD=USR(ADR(ML$),ADR(A$),AD)                <XM>
520 ON CD GOSUB 590,690,710                    <PH>
530 AD=AD+8*(CD=0)                             <SV>
540 IF AD<=L THEN 510                          <WK>
550 AD=L:GOSUB 590                             <HI>
560 POKE 16,192:POKE 53774,192:END             <XR>
590 TRAP 600:CLOSE #1:XIO 33,#1,0,0,FB$        <MD>
600 TRAP 610:CLOSE #1:XIO 32,#1,0,0,FR$        <IF>
610 TRAP 660:CLOSE #1:OPEN #1,8,0,F$           <MU>
620 E=USR(ADR(CIO$),1,11,ADR(A$),AD)           <LC>
630 CLOSE #1                                   <LI>
640 IF E<>1 THEN ? "Disk-Error▲";E             <ZN>
650 TRAP 65535:RETURN                          <FB>
660 CLOSE #1                                   <LO>
670 ? "Disk-Error▲";PEEK(195):RETURN           <XX>
690 ? CHR$(253):? "Pruefsummenfehler"          <PO>
700 RETURN                                     <MR>
710 ? "Fehler!!!!";:LIST 510:END               <CW>
1000 Q=0                                       <HO>
1010 FOR I=1536 TO 1592                        <NU>
1020 READ A:Q=Q+A:POKE I,A                     <KX>
1030 NEXT I                                    <EY>
1040 IF Q<>4196 THEN ? "Data▲Error":LIST
 1300,1390:END                                 <JV>
1050 Q=0                                       <XD>
1060 FOR I=1 TO 261                            <RZ>
1070 READ A                                    <QI>
1080 IF A>255 THEN GOSUB 1190                  <EL>
1090 ML$(I)=CHR$(A):Q=Q+A                      <MU>
1100 NEXT I                                    <ER>
1110 READ A:IF Q<>A THEN 1200                  <FZ>
1120 Q=0                                       <WW>
1130 FOR I=1 TO 83                             <JT>
1140 READ A:Q=Q+A:CIO$(I)=CHR$(A)              <QL>
1150 NEXT I                                    <FG>
1160 IF Q<>10894 THEN ? "Data▲Error▲";:L
IST 1950,2080:END                              <MP>
1170 RETURN                                    <PQ>
1190 IF Q=A THEN READ A:Q=0:RETURN             <VA>
1200 ? "Data▲Error"                            <GO>
1210 IF A=7931 THEN LIST 1430,1530             <IU>
1220 IF A=6585 THEN LIST 1540,1640             <PX>
1230 IF A=8135 THEN LIST 1650,1750             <RD>
1240 IF A=7184 THEN LIST 1760,1860             <XK>
1250 IF A=2222 THEN LIST 1870,1910             <OU>
```

```
1300 DATA▲173,37,228,72,173,36                 <YX>
1310 DATA▲228,72,96,72,74,74                   <WJ>
1320 DATA▲74,74,32,20,6,104                    <ZA>
1330 DATA▲41,15,9,48,201,58                    <JO>
1340 DATA▲144,2,105,6,168,173                  <OH>
1350 DATA▲71,3,72,173,70,3                     <DW>
1360 DATA▲72,152,96,50,31,30                   <CG>
1370 DATA▲26,24,29,27,51,53                    <IL>
1380 DATA▲48,63,21,18,58,42                    <KV>
1390 DATA▲56,190,52                            <PY>
1430 DATA▲169,0,133,213,104,201                <HX>
1440 DATA▲2,240,16,170,240,5                   <YR>
1450 DATA▲104,104,202,208,251,169              <RJ>
1460 DATA▲3,44,169,1,133,212                   <BU>
1470 DATA▲96,104,133,215,104,133               <YK>
1480 DATA▲214,104,133,217,104,133              <GO>
1490 DATA▲216,24,101,214,133,214               <UF>
1500 DATA▲165,217,101,215,133,215              <IU>
1510 DATA▲169,155,32,28,6,165                  <SR>
1520 DATA▲217,32,9,6,165,216                   <LG>
1530 DATA▲7931                                 <MS>
1540 DATA▲32,9,6,169,58,32                     <OX>
1550 DATA▲28,6,169,0,133,218                   <JQ>
1560 DATA▲173,252,2,41,191,162                 <NS>
1570 DATA▲17,221,39,6,240,5                    <DP>
1580 DATA▲202,16,248,48,239,134                <GQ>
1590 DATA▲219,141,252,2,32,0                   <UW>
1600 DATA▲6,41,127,32,28,6                     <GB>
1610 DATA▲165,219,201,16,240,168               <IB>
1620 DATA▲144,40,165,218,240,205               <CF>
1630 DATA▲74,176,5,169,126,32                  <RS>
1640 DATA▲6585                                 <PH>
1650 DATA▲28,6,160,4,70,220                    <YY>
1660 DATA▲102,221,102,222,102,223              <VE>
1670 DATA▲102,224,102,225,102,226              <DY>
1680 DATA▲102,227,102,228,136,208              <QR>
1690 DATA▲235,198,218,184,80,178               <BA>
1700 DATA▲10,10,10,10,160,4                    <LO>
1710 DATA▲10,38,228,38,227,38                  <UU>
1720 DATA▲226,38,225,38,224,38                 <YP>
1730 DATA▲223,38,222,38,221,38                 <RU>
1740 DATA▲220,136,208,234,230,218              <OG>
1750 DATA▲8135                                 <LZ>
1760 DATA▲165,218,201,18,240,19                <WO>
1770 DATA▲201,16,208,4,169,60                  <MK>
1780 DATA▲208,5,74,176,206,169                 <EG>
1790 DATA▲32,32,28,6,184,80                    <JZ>
1800 DATA▲198,165,217,10,101,216               <ZP>
1810 DATA▲10,101,220,10,101,221                <UO>
1820 DATA▲10,101,222,10,101,223                <YP>
1830 DATA▲10,101,224,10,101,225                <CQ>
1840 DATA▲10,101,226,10,101,227                <GR>
1850 DATA▲197,228,208,18,169,62                <NJ>
1860 DATA▲7184                                 <NS>
1870 DATA▲32,28,6,160,7,185                    <LL>
1880 DATA▲220,0,145,214,136,16                 <KJ>
1890 DATA▲248,169,0,44,169,2                   <HU>
1900 DATA▲133,212,96                           <CA>
1910 DATA▲2222                                 <HM>
1950 DATA▲104,201,4,208,66,104                 <KE>
1960 DATA▲104,170,104,104,133,212              <AX>
1970 DATA▲104,133,214,104,133,213              <DH>
1980 DATA▲104,133,216,104,133,215              <HO>
1990 DATA▲160,134,138,201,8,176                <AX>
2000 DATA▲33,10,10,10,10,170                   <LS>
2010 DATA▲165,212,157,66,3,165                 <VR>
2020 DATA▲213,157,68,3,165,214                 <TO>
2030 DATA▲157,69,3,165,215,157                 <AD>
2040 DATA▲72,3,165,216,157,73                  <SE>
2050 DATA▲3,32,86,228,132,212                  <HM>
2060 DATA▲169,0,133,213,96,160                 <QU>
2070 DATA▲10,170,240,244,104,104               <SB>
2080 DATA▲202,208,251,240,237                  <JQ>
```

Eingabehilfe »AMPEL« für Programme in Maschinensprache. Bitte mit dem Prüfsummer von Seite 60 eingeben.

Auf die Kassette, fertig, los!

Benutzer eines Kassettenrecorders dürfen endlich aufatmen. Auch sie können nun die Vorteile des »Turbo-Basic« voll nutzen.

Alle (Atari-)Welt spricht vom Turbo-Basic und erfreut sich der immensen Vorteile und erweiterten Fähigkeiten gegenüber dem »Atari-Basic« – vorausgesetzt, man verfügt über ein Diskettenlaufwerk. Wer indes nur einen Kassettenrecorder sein eigen nennt, stand bislang im Regen. Die Ladeprozedur des Turbo-Basic war nämlich ausschließlich auf Diskettenbetrieb ausgelegt. Diesem Mißstand macht das Programm »Turbo-Load« jetzt ein Ende. Sollten Sie der Vorzüge des Turbo-Basic noch nicht gewahr sein, hier ein paar Appetitanreger:

Der Turbo-Basic-Interpreter

- ist voll kompatibel zum Atari-Basic
- macht auch Atari-Basic-Programme wesentlich schneller
- erweitert den Befehlssatz um so wichtige Elemente wie neue Schleifenformen (DO-UNTIL, WHILE-WEND etc.) und Grafik-Kommandos (beispielsweise zum Zeichnen von Kreisen und Rechtecken sowie zum Füllen beliebiger Flächen).

Eine vollständige Aufstellung des Befehlsvorrats und der Handhabung des Interpreters entnehmen Sie bitte dem Beitrag »Basic, schnell wie der Wind« ab Seite 68 dieser Ausgabe.

Geben Sie zunächst das Turbo-Load-Listing mit AMPEL ein und speichern Sie es. Auf der gleichen Kassette, genau hinter Turbo-Load, speichern Sie dann das ebenfalls eingegebene Turbo-Basic. Zum Schutz vor unwiederbringlichen Datenverlusten fertigen Sie mindestens eine Arbeitskopie von dieser Turbo-Basic-Kassette. Zum Einsatz des Turbo-Basic spulen Sie nun die Kassette auf den Anfang von Turbo-Load und schalten den Computer ein, während Sie gleichzeitig die Tasten <OPTION> und <START> gedrückt halten. Hören Sie den Summton, lassen Sie die Tasten wieder los und starten den Recorder. Nach einer kurzen Ladezeit fordert Turbo-Load Sie auf, die Kassette an den Beginn der Turbo-Basic-Datei zu spulen, die Taste <PLAY> des Recorders und abschließend <RETURN> auf der Computer-Tastatur zu drücken. Eine knappe Viertelstunde später meldet sich Turbo-Basic zur Stelle.

Der Interpreter ist fest im Computer installiert, auch die <RESET>-Taste bringt ihn nicht aus der Ruhe. Geben Sie den Befehl »? FRE(0)« ein. Wenn der Computer die Zahl »34018« ausgibt, ist alles in Ordnung. (ja)

Steckbrief	
Programm:	Turbo-Load
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Kassette

```
Programmname :D:TURBOLOA.BIN
Laenge       :904 Bytes

0000:00 08 00 07 21 07 A9 3C <24>
0008:8D 02 D3 A9 21 85 0C A9 <14>
0010:07 85 0D A9 22 85 0A A9 <01>
0018:07 85 0B 18 60 00 00 00 <DA>
0020:00 60 A2 00 BD 00 E0 9D <AF>
0028:00 0C BD 00 E1 9D 00 0D <49>
0030:BD 00 E3 9D 00 0F E8 D0 <90>
0038:EB A2 00 BD 36 09 9D 00 <F0>
0040:0E E8 E0 88 D0 F5 A9 0C <87>
0048:8D C5 02 A9 10 8D C6 02 <FA>
0050:8D F0 02 A9 0C 8D F4 02 <EB>
0058:A2 00 A9 0B 8D 42 03 A9 <27>
0060:BE 8D 44 03 A9 09 8D 45 <E4>
0068:03 A9 A7 8D 48 03 A9 00 <51>
0070:8D 49 03 20 56 E4 A2 10 <C0>
0078:A9 03 9D 42 03 A9 85 9D <A8>
0080:44 03 A9 0A 9D 45 03 A9 <EA>
0088:04 9D 4A 03 A9 80 9D 4B <38>
0090:03 20 56 E4 A2 10 A9 07 <AA>
0098:9D 42 03 A9 1D 9D 44 03 <C9>
00A0:A9 07 9D 45 03 A9 02 9D <7A>
00A8:48 03 A9 00 9D 49 03 20 <D6>
00B0:56 E4 10 0A C0 88 F0 03 <7A>
00B8:4C 3E 08 4C 68 08 A9 FF <D7>
00C0:CD 1D 07 D0 05 CD 1E 07 <EE>
00C8:F0 CA AD 1E 07 C9 C0 90 <57>
00D0:09 38 AD 1E 07 E9 50 8D <04>
00D8:1E 07 A2 10 A9 1F 9D 44 <3F>
00E0:03 A9 07 9D 45 03 20 56 <FB>
00E8:E4 10 03 4C 3E 08 AD 20 <81>
00F0:07 C9 C0 90 09 38 AD 20 <56>
00F8:07 E9 50 8D 20 07 AD 1F <10>
0100:07 38 ED 1D 07 8D 1F 07 <B2>
0108:AD 20 07 ED 1E 07 8D 20 <DE>
0110:07 EE 1F 07 D0 03 EE 20 <EB>
0118:07 A2 10 AD 1D 07 9D 44 <55>
0120:03 AD 1E 07 9D 45 03 AD <B4>
0128:1F 07 9D 48 03 AD 20 07 <95>
0130:9D 49 03 20 56 E4 30 03 <DA>
0138:4C 94 07 4C 3E 08 20 5D <9A>
0140:08 A2 00 A9 65 8D 44 03 <66>
0148:A9 0A 8D 45 03 A9 20 8D <AD>
0150:48 03 A9 00 8D 49 03 20 <FB>
0158:56 E4 4C 22 07 A2 10 A9 <51>
0160:0C 9D 42 03 20 56 E4 60 <8A>
0168:AE 24 E4 AC 25 E4 E8 D0 <DA>
0170:01 C8 8E F8 24 8C F9 24 <9B>
0178:A9 00 8D 0E D4 78 A9 FE <15>
0180:8D 01 D3 A9 AB 8D FA FF <FA>
0188:A9 24 8D FB FF A9 D0 8D <91>
0190:FE FF A9 24 8D FF FF A9 <B8>
0198:CC 20 AD 08 A9 E0 20 AD <74>
01A0:08 A9 40 8D 0E D4 58 20 <30>
01A8:D3 08 4C 47 60 85 D5 A0 <62>
01B0:00 84 D4 A2 04 A9 FF 8D <79>
01B8:01 D3 B1 D4 99 00 21 C8 <12>
01C0:D0 F8 CE 01 D3 B9 00 21 <10>
01C8:91 D4 C8 D0 F8 E6 D5 CA <D9>
01D0:D0 E3 60 A9 FE 8D 01 D3 <01>
01D8:A9 00 85 D6 A9 70 85 D7 <22>
01E0:A9 F7 85 D8 A9 7B 85 D9 <1A>
01E8:20 12 09 A9 00 85 D6 A9 <81>
01F0:88 85 D7 A9 F9 85 D8 A9 <83>
01F8:8F 85 D9 20 12 09 A9 00 <D6>
0200:85 D6 A9 94 85 D7 A9 F1 <A6>
0208:85 D8 A9 AF 85 D9 20 12 <BE>
0210:09 60 A5 D7 18 69 50 85 <3A>
0218:DB A0 00 84 DA B1 D6 91 <CC>
0220:DA C4 D8 D0 06 A5 D7 C5 <4F>
0228:D9 F0 0A C8 D0 EF E6 D7 <2F>
0230:E6 DB 4C 1D 09 60 00 18 <FA>
0238:18 FF 3C 3C 66 42 00 00 <7F>
0240:00 06 66 7E 76 66 00 00 <C7>
0248:18 3C 66 7E 66 66 00 3E <22>
0250:73 33 3E 30 30 78 00 7C <E0>
0258:37 33 3E 30 30 78 00 00 <AA>
0260:60 63 36 3C 39 63 00 01 <C7>
0268:01 0F 03 03 F6 E4 00 80 <00>
0270:80 F0 C0 C0 60 20 00 01 <A8>
0278:03 06 06 06 03 01 00 E1 <7D>
0280:33 06 06 06 33 E1 00 C7 <84>
0288:63 33 36 36 66 CE 00 0E <69>
0290:9C 9C F6 66 66 67 00 7E <35>
0298:33 33 3E 30 30 78 00 67 <53>
02A0:66 66 66 66 66 3C 00 FF <B3>
02A8:31 31 31 31 31 7B 00 FB <3D>
02B0:81 81 F1 81 81 FB 00 F8 <BB>
02B8:8C 8C F8 B0 98 CC 7D 9B <33>
02C0:20 20 20 20 20 20 20 20 <A0>
02C8:20 20 20 20 20 20 00 01 <69>
02D0:02 03 04 05 06 07 9B 20 <08>
02D8:20 20 20 20 20 20 20 20 <D8>
02E0:20 20 20 20 08 09 0A 0B <66>
02E8:0C 0D 0E 0F 10 9B 9B 20 <2D>
02F0:20 20 20 20 20 20 20 20 <70>
02F8:20 20 54 55 52 42 4F 20 <BC>
0300:4C 4F 41 44 20 31 2E 30 <9E>
0308:9B 20 20 20 20 20 20 20 <CB>
0310:28 63 29 20 31 39 38 37 <FB>
0318:20 20 48 65 6E 72 69 6B <F8>
0320:20 46 69 73 63 68 9B 9B <76>
0328:9B 20 20 20 20 20 20 20 <AB>
0330:20 20 54 55 52 42 4F 20 <7E>
0338:42 41 53 49 43 20 58 4C <3D>
0340:20 31 2E 35 9B 20 20 20 <71>
0348:20 20 20 28 63 29 20 31 <15>
0350:39 38 35 20 46 72 61 6E <90>
0358:6B 20 4F 73 74 72 6F 77 <57>
0360:73 6B 69 9B 9B 9B 46 65 <6A>
0368:68 6C 65 72 20 62 65 69 <9D>
0370:6D 20 4C 61 64 65 6E 20 <F5>
0378:76 6F 6E 20 4B 61 73 73 <BA>
0380:65 74 74 65 21 43 00 00 <5F>
```

Listing. »Turbo-Load«

Wenn die Kassette mit der Diskette

Mit dem hier vorgestellten Programm lassen sich alle für Diskette gedachten Maschinenprogramme von Kassette laden und starten.

Für uns ist das schon ein Problem, die Sache mit den Programmen auf Diskette. Fast ausnahmslos alle Programme, die wir im Heft abdrucken, bekommen wir auf Diskette zugesandt. Das allein wäre nicht weiter schlimm. Nur, meistens sind die Programme so ausgelegt, daß sie nur mit der Diskettenstation zusammenarbeiten. Ganz schlimm wird es, wenn wir Maschinenprogramme erhalten. Es ist fast unmöglich, die Programme so umzuschreiben, daß sie sowohl mit Kassette als auch mit Diskette funktionieren. Außerdem müßten wir dann immer zwei Versionen abdrucken, einmal für Kassetten- und einmal für Diskettenbetrieb. Da wir den Platz im Heft lieber mit anderen Programmen oder Beiträgen füllen wollen, haben wir uns eine elegante Lösung einfallen lassen, die Nachteile weder für den Kassetten- noch für den Diskettenbetrieb mit sich bringt.

Tippen Sie das Programm mit AMPEL ab und speichern Sie es auf Ihrer Kassette. Unser »COS«, »Cassetten Operations System«, wird geladen, indem Sie den Computer ausschalten, die <START>-Taste drücken, je nachdem, ob Sie Basic benötigen oder nicht, die <OPTION>-Taste betätigen und den Computer dann einschalten. Er meldet sich mit einem Hupton. Spulen Sie jetzt die Kassette auf den Anfang von COS, schalten den Recorder auf »PLAY« und drücken eine Taste am Computer. COS wird nun geladen. Das Programm beansprucht acht Blöcke auf der Kassette und ist somit schnell im Speicher. COS meldet sich mit einem Menü.

Drei Funktionen hat COS. Die wichtigste ist die Funktion <L>, mit der ein File von der Kassette geladen und gestartet wird. Drücken Sie diese Taste, fordert der Computer Sie mit einem Hupton auf, die Kassette auf den Anfang des zu ladenden Files zu spulen, ihn wieder auf »PLAY« zu schalten und eine Taste zu betätigen. »Turbo-Basic« läßt sich mit COS leider nicht laden, da Turbo-Basic eine spezielle Laderoutine für den Diskettenbetrieb eingebaut hat, die mit Kassette nicht funktioniert.

Mit der zweiten Funktion <C> wie »Cartridge« wird zu einem eingesteckten Modul gesprungen beziehungsweise das Basic gestartet. Verwenden Sie diese Funktion nicht, wenn Sie Basic nicht eingeschaltet haben. Der Computer hängt sich sonst auf, und Sie müssen ihn mit <RESET> wieder ins Diesseits der Bits und Bytes zurückrufen. Diese Funktion wird auch zusammen mit dem Prüfsummer verwendet. Nachdem Sie den Prüfsummer von Kassette geladen haben, müssen Sie irgendwie ins Basic kommen, um den Prüfsummer benutzen zu können. Dies geschieht mit der Funktion <C>.

Als letztes steht die Funktion <S> zur Verfügung, mit der ein geladenes Programm nochmals gestartet wird. Nützlich ist dies bei verschiedenen kleinen Demonstrationsprogrammen, die nach getaner Arbeit wieder ins COS zurückkehren. (hf)

Steckbrief	
Programm:	COS
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Kassette

```
Programmname :COS.BIN
Länge        :808 Bytes
---------------------------------

0000:00 07 00 07 3D 07 A9 3C <C7>
0008:8D 02 D3 A9 3D 85 0C A9 <F1>
0010:07 85 0D A9 48 85 0A A9 <32>
0018:07 85 08 A9 00 8D 44 02 <A9>
0020:A9 47 8D 38 07 8D 36 07 <99>
0028:A9 07 8D 39 07 8D 37 07 <C7>
0030:18 60 00 00 00 00 00 00 <14>
0038:00 00 00 00 00 A9 21 8D <AE>
0040:E7 02 A9 0A 8D E8 02 60 <59>
0048:A9 01 85 09 A2 00 A9 0B <67>
0050:8D 42 03 8D F0 02 A9 48 <79>
0058:8D 44 03 A9 09 8D 45 03 <B2>
0060:A9 99 8D 48 03 A9 00 8D <2C>
0068:49 03 20 56 E4 20 0A 09 <73>
0070:AD 3A 07 C9 4C F0 1F C9 <EC>
0078:53 D0 03 4C 96 08 C9 43 <64>
0080:D0 EB A2 00 8E F0 02 8A <B2>
0088:8D 48 03 8D 49 03 A9 7D <BF>
0090:20 56 E4 6C FE BF A9 47 <FE>
0098:8D 38 07 8D 36 07 A9 07 <35>
00A0:8D 39 07 8D 37 07 A9 00 <8E>
00A8:8D 3B 07 8D 3C 07 A2 10 <31>
00B0:A9 03 9D 42 03 A9 1F 9D <33>
00B8:44 03 A9 0A 9D 45 03 A9 <02>
00C0:04 9D 4A 03 A9 80 9D 4B <6F>
00C8:03 20 56 E4 A9 47 8D E0 <B6>
00D0:02 A9 07 8D E1 02 A2 10 <48>
00D8:A9 07 9D 42 03 A9 32 9D <62>
00E0:44 03 A9 07 9D 45 03 A9 <99>
00E8:02 9D 48 03 A9 00 9D 49 <19>
00F0:03 20 56 E4 10 0A C0 88 <11>
00F8:D0 03 4C 8A 08 4C C1 08 <BE>
0100:A9 FF CD 32 07 D0 0B CD <AD>
0108:33 07 D0 06 8D 3C 07 4C <8F>
0110:D6 07 AD 3C 07 D0 03 4C <62>
0118:E8 08 AD 3B 07 D0 0C AD <95>
0120:E2 02 8D 36 07 AD E3 02 <3F>
0128:8D 37 07 A2 10 A9 07 8D <77>
0130:3B 07 9D 42 03 A9 34 9D <09>
0138:44 03 A9 07 9D 45 03 A9 <4F>
0140:02 9D 48 03 A9 00 9D 49 <AB>
0148:03 20 56 E4 30 73 EE 34 <38>
0150:07 D0 03 EE 35 07 A2 10 <22>
0158:A9 07 9D 42 03 AD 32 07 <62>
0160:9D 44 03 AD 33 07 9D 45 <F2>
0168:03 AD 34 07 38 ED 32 07 <EE>
0170:9D 48 03 AD 35 07 ED 33 <E0>
0178:07 9D 49 03 20 56 E4 30 <CC>
0180:40 20 87 08 4C CC 07 6C <27>
0188:E0 02 AD E2 02 8D 38 07 <F8>
0190:AD E3 02 8D 39 07 20 B6 <17>
0198:08 20 B0 08 A9 47 CD E2 <1B>
01A0:02 D0 0A A9 07 CD E3 02 <96>
01A8:D0 03 20 B3 08 4C 48 07 <89>
01B0:6C 38 07 6C 36 07 A2 10 <B9>
01B8:A9 0C 9D 42 03 20 56 E4 <F3>
01C0:60 20 B6 08 A2 00 A9 0B <A3>
01C8:8D 42 03 A9 E1 8D 44 03 <5F>
01D0:A9 09 8D 45 03 A9 21 8D <F9>
01D8:48 03 A9 00 8D 49 03 20 <84>
01E0:56 E4 20 0A 09 4C 48 07 <38>
01E8:20 B6 08 A2 00 A9 02 8D <B7>
01F0:44 03 A9 0A 8D 45 03 A9 <37>
01F8:1D 8D 48 03 A9 00 8D 49 <4D>
0200:03 20 56 E4 20 0A 09 4C <1D>
0208:48 07 A2 20 A9 03 9D 42 <FC>
0210:03 A9 20 9D 44 03 A9 0A <47>
0218:9D 45 03 A9 04 9D 4A 03 <4F>
0220:20 56 E4 A9 07 9D 42 03 <C6>
0228:A9 3A 9D 44 03 A9 07 9D <4C>
0230:45 03 A9 01 9D 48 03 A9 <71>
0238:00 9D 49 03 20 56 E4 A9 <87>
0240:0C 9D 42 03 20 56 E4 60 <A8>
0248:7D 48 61 70 70 79 2D 43 <AA>
0250:6F 6D 70 75 74 65 72 20 <84>
0258:43 4F 53 20 20 20 41 54 <54>
0260:41 52 49 20 56 65 72 73 <C0>
0268:69 6F 6E 20 31 2E 30 43 <11>
0270:6F 70 79 72 69 67 68 74 <03>
0278:20 28 63 29 20 31 39 38 <65>
0280:37 20 62 79 20 48 65 6E <66>
0288:72 69 6B 20 46 69 73 63 <A7>
0290:68 9B 9B 5B 4C 5D 20 46 <12>
0298:69 6C 65 20 76 6F 6E 20 <85>
02A0:43 61 73 73 65 74 74 65 <5A>
02A8:20 6C 61 64 65 6E 9B 5B <5F>
02B0:53 5D 20 47 65 6C 61 64 <29>
02B8:65 6E 65 6E 20 46 69 6C <B3>
02C0:65 20 73 74 61 72 74 65 <1D>
02C8:6E 9B 5B 43 5D 20 47 65 <67>
02D0:68 65 20 7A 75 72 20 43 <7E>
02D8:61 72 74 72 69 64 67 65 <63>
02E0:9B 9B 9B 46 65 68 6C 65 <85>
02E8:72 20 62 65 69 6D 20 4C <C9>
02F0:61 64 65 6E 20 76 6F 6E <C5>
02F8:20 4B 61 73 73 65 74 74 <1E>
0300:65 9B 9B 9B 46 65 68 6C <6D>
0308:65 72 20 2D 20 6B 65 69 <F6>
0310:6E 20 44 4F 53 20 32 2E <6E>
0318:30 20 46 69 6C 65 9B 43 <E8>
0320:4B 00 00 00 00 00 00 00 <92>
```

Listing. »COS«, mit AMPEL eingeben

Basic, schnell wie der Wind – mit dem Turbo-Basic-XL-Interpreter

Ist Ihnen Atari-Basic zu langsam? Oder wünschen Sie sich zusätzliche Befehle? Dann ist unser Turbo-Basic-Interpreter genau das richtige Programm für Sie.

Zugegeben, unser Interpreter ist ein wenig umfangreich. Immerhin müssen insgesamt 18108 Byte erst einmal eingetippt sein. Eine Aufgabe, die sicher einige Stunden in Anspruch nimmt. Haben Sie aber dann den Interpreter auf Diskette vorliegen, sparen Sie sich zukünftig viel Zeit für die Entwicklung neuer Programme. Erstens stehen Ihnen sämtliche Befehle des normalen Atari-Basic zur Verfügung, und zweitens gewinnen Sie durch die neuen Befehle viel Zeit und vor allem Speicherplatz.

Aber Turbo-Basic XL spart nicht nur Zeit bei der Programmierung, sondern auch bei der Ausführung von Programmen. Es macht seinem Namen als schnellstes derzeit erhältliches Basic für Atari-Computer nämlich alle Ehre. Wir haben unseren Interpreter dem normalen Basic und verschiedenen Basic-Compilern gegenübergestellt. Turbo-Basic XL siegte in jeder Hinsicht.

Damit Sie mit Turbo-Basic XL und dem dazugehörigen Compiler arbeiten können, benötigen Sie unbedingt einen Atari 800XL oder 130XE. Besitzer der älteren Computer, also Atari 400 und 800, können den Interpreter leider nicht benutzen. In der abgedruckten Version setzt Turbo-Basic XL mindestens 64 KByte RAM voraus.

Beachten Sie bitte: Turbo-Basic XL muß unbedingt mit AMPEL (Atari-Maschinen-Programm-Eingabe-Listing) eingegeben werden. Von Basic aus läßt sich Turbo-Basic XL nämlich nicht eingeben. Wenn Sie alle dazugehörigen Erklärungen befolgen, kann nichts schiefgehen.

Besondere Hinweise zu Turbo-Basic XL

Turbo-Basic XL ist als »AUTORUN.SYS«-File ausgelegt. Das heißt, wenn Sie den Interpreter komplett eingegeben haben, können Sie Ihrem File einfach den Namen »AUTORUN.SYS« geben. Nach dem Einschalten des Computers wird dann automatisch Turbo-Basic XL geladen. Anschließend können Sie auch schon mit dem Interpreter arbeiten. Einzige Voraussetzung: Die Files »DOS.SYS« und »DUP.SYS« (entweder DOS 2.0 oder 2.5) müssen sich auf der gleichen Diskette befinden. Mit DOS 3.0 arbeitet Turbo-Basic XL übrigens nicht!

Hier noch einige Hinweise, wie Sie einen Dateinamen auf Diskette ändern können.

Nehmen wir an, Sie haben Ihrem Interpreter bei der Eingabe mit AMPEL den Namen »TURBO« gegeben. Dieses File soll nun den Namen »AUTORUN.SYS« erhalten. Entfernen Sie also zunächst Ihre Turbo-Basic-XL-Diskette und schalten Sie bitte den Computer aus. Nachdem Sie eine formatierte Diskette, auf der sich die Files »DOS.SYS« und »DUP.SYS« befinden, ins Laufwerk gelegt haben, schalten Sie den Computer wieder ein. Anschließend geben Sie »DOS« ein und gelangen über die <RETURN>-Taste ins DOS-Menü. Nun rufen Sie auf der Turbo-Basic-XL-Diskette mit »E« die Funktion »Rename File« auf und drücken <RETURN>. Die Meldung »Give – Old Name, New Name« erscheint. Geben Sie beispielsweise »TURBO,AUTORUN.SYS« ein. Sollte Ihre Datei einen anderen Namen haben, müssen Sie diesen anstelle von »TURBO« einsetzen. Nach Betätigung der <RETURN>-Taste wird Ihr File umbenannt und hat dann den Namen »AUTORUN.SYS«. Überzeugen Sie sich, daß der soeben beschriebene Vorgang auch wirklich funktioniert hat. Mit »A« für Directory und <RETURN> wird das neue Directory auf dem Bildschirm angezeigt. Folgende Programme sollten sich jetzt auf Ihrer Diskette befinden: DOS.SYS, DUP.SYS und AUTORUN.SYS.

Noch eins, bevor Sie mit Turbo-Basic XL arbeiten: Sie sollten sich mindestens eine Sicherheitskopie Ihrer Diskette anlegen. Nehmen Sie sich also eine neue oder eine Diskette, die Sie löschen können, und formatieren Sie diese. Falls Sie über DOS 2.5 verfügen, können Sie sich das Formatieren sparen. Mit der Funktion »J« im DOS-Menü wird nun die Diskette kopiert. Wenn Sie ein Diskettenlaufwerk besitzen, geben Sie jetzt »D1,D1« ein und betätigen daraufhin die <RETURN>-Taste. Der Kopiervorgang beginnt. Dabei werden Sie mehrmals aufgefordert, die Original-Diskette mit der Duplikat-Diskette auszutauschen.

Sie haben jetzt also zwei Disketten vorliegen, die exakt die gleichen Programme enthalten. Arbeiten Sie aber bitte nur mit einer der Disketten und legen Sie die andere an eine sichere Stelle.

Nach dieser Sicherheitsmaßnahme können Sie endlich Turbo-Basic XL ausprobieren. Dazu muß sich nur Ihre Interpreter-Diskette im Laufwerk befinden. Schalten Sie jetzt die Stromversorgung Ihres Computers aus und wieder ein. Wenn Sie dabei keine Taste, auch nicht die

```
250 END
999 ------------------------------
1000 PROC RECHNEN
1010   GRAPHICS 15+16:POKE 764,255
1030   DX=(XMAX-XMIN)/159
1040   DY=(YMAX-YMIN)/191
1050   CX=XMIN:CY=YMAX
1080   FOR ZEILE=0 TO 191
1090     FOR SPALTE=0 TO 159
1110       TI=0:XW=0:YW=0:X2=0:Y2=0
1120       WHILE TI<TMAX AND (X2+Y2)<8
1130         YW=2*XW*YW-CY:XW=X2-Y2-CX
1150         X2=XW^2:Y2=YW^2:TI=TI+1
1180       WEND
1190       IF TI=TMAX
1200         COLOR %0
1220       ELSE
1230         COLOR (TI MOD 3)+%1
1240       ENDIF
1250       CX=CX+DX:PLOT SPALTE,ZEILE
1260       IF PEEK(764)=28 THEN POP :GOT
0 1295
1270     NEXT SPALTE
1280     CX=XMIN:CY
```

Ein typischer Turbo-Basic-XL-Bildschirm

<OPTION>-Taste, mit der normalerweise das eingebaute Basic abgestellt wird, betätigt haben, erscheint das Titelbild und kurz danach die READY-Meldung. Sie befinden sich jetzt in Turbo-Basic XL. Viel Spaß beim Programmieren!

Strukturierte Programmierung

In der folgenden Beschreibung bedeuten:
aexp=arithmetic expression (arithmetischer Ausdruck)
sexp=string expression (Text Ausdruck (A$, "TEXT", CHR$(), STR$(), HEX$())
lineno=Zeilennummer
»...«=Ein oder mehrere Befehle. In einer IF-THEN-Abfrage handelt es sich hierbei um den nach THEN folgenden Text. Sonst gibt es keine Beschränkung. »...« kann also auch mehrere Programmzeilen und/oder den Teil einer Programmzeile umfassen.

Es gibt im Standard-Atari-Basic nur zeilennummerorientierte Sprungbefehle und die »FOR...NEXT«-Schleife zur Konstruktion von Schleifen. In Turbo-Basic gibt es zusätzliche, an Pascal angelehnte Strukturelemente.
IF aexp THEN lineno
IF aexp THEN ...
Das normale IF...THEN Statement.

IF aexp ... **ENDIF**
IF aexp ... **ELSE** ... **ENDIF**

Wenn die Bedingung aexp erfüllt (< >0) ist, wird der Programmteil zwischen IF und ELSE, sonst der zwischen ELSE und ENDIF ausgeführt. ELSE kann auch entfallen. Als Trennzeichen nach aexp dient nicht THEN, sondern ein Doppelpunkt (:) oder Zeilenende (RETURN). Auch vor und nach ELSE und ENDIF muß ein solches Trennzeichen stehen. Nach ELSE darf keine Zeilennummer folgen. In Atari-Basic benötigt man dafür oft mehrere GOTO-Befehle.

REPEAT ... **UNTIL** aexp

Wiederholt die Anweisungen »...«, bis die Bedingung erfüllt ist. Hier wird die Bedingung erst am Ende der Schleife geprüft. Der Programmteil zwischen REPEAT und UNTIL wird also mindestens einmal ausgeführt.

WHILE aexp ... **WEND**

Wiederholt »...«, solange die Bedingung erfüllt ist, das heißt, wenn die Bedingung das erste Mal nicht erfüllt ist, wird die Schleife kein einziges Mal ausgeführt.

DO ... LOOP

Endlosschleife. Wiederholt die Anweisungen »...« immer wieder von neuem.

EXIT

Verläßt eine Schleife, Sprung ans Schleifenende. Dieser Befehl ist verwendbar bei »DO ... LOOP« und »REPEAT ... UNTIL«, »WHILE ... WEND« und auch bei »FOR ... NEXT«-Schleifen. Dieser Befehl stellt eine Art Notausgang aus Schleifen dar, bei »DO ... LOOP« sogar den einzigen, der in strukturierten Programmen erlaubt ist. Es wird stets an das Ende der Schleife gesprungen. Eine Schleife läßt sich auch mit »POP:GOTO lineno« abbrechen. Dies sollte jedoch nur im Notfall geschehen, da dieser Befehl nicht zur übersichtlichen Programmierung beiträgt.

★F
★F +

Nach diesem Befehl sind »FOR ... NEXT«-Schleifen abweisend. Es wird also vor dem ersten Durchlauf der Schleife geprüft, ob der Zähler schon den Endwert erreicht hat. Dazu ein Beispiel: »FOR I=2 TO 1:? I:NEXT I«.

1. Der Anfangswert der Schleife ist 2
2. Schleifenzähler um 1 erhöhen
3. Vergleich mit dem Endwert (1)
4. Verlassen der Schleife, da die Bedingung erfüllt ist.

Nach ★F + wird zuerst die Variable I mit dem Anfangswert 2 geladen und dann mit der Variablen I (Endwert=1) verglichen. Anschließend wird die Schleife bis zum »NEXT I«-Befehl übersprungen. Es erfolgt keine Ausgabe auf dem Bildschirm. Der »F+«-Befehl erspart oft eine spezielle Abfrageroutine.

★F -

Stellt den Normalzustand wieder her. »FOR ... NEXT«-Schleifen werden mindestens einmal durchlaufen. Auch bei RUN wird automatisch ein »★F -« ausgeführt (entspricht dem Atari-Basic).

PROC name

Beginn eines Unterprogramms (Prozedur) mit dem Namen name.

ENDPROC

Prozedurende. Entspricht dem RETURN nach einem »GOSUB lineno«.

EXEC name

Ruft die Prozedur name auf, entspricht »GOSUB lineno«. Die normalen Befehle GOSUB und RETURN erlauben, nur Unterprogramme mit einer Zeilennummer aufzurufen. Hier ist der Aufruf mit einem leichter zu behaltenden Namen möglich. Außerdem belegt EXEC-PROC-ENDPROC meist sogar weniger Speicherplatz als GOSUB-RETURN und ist etwas schneller. Die Prozedurnamen werden genauso gespeichert wie die Variablennamen. Jeder Name belegt 8 Byte plus ein Byte für jedes Zeichen. Jede weitere Verwendung benötigt dann nur ein (zwei) Byte. Die Zeilennummer einer GOSUB-Anweisung belegt dagegen 7 Byte. Obwohl Turbo-Basic XL kompatibel zu Atari-Basic ist, können statt 128 jetzt 256 verschiedene Variablen- oder Prozedurnamen verwendet werden. Ab der 129sten Variablen kostet jede Verwendung zwei statt bisher nur 1 Byte Speicherplatz.

ON aexp **EXEC** pname,pnam...

Entspricht der Atari-Basic-Anweisung »ON GOSUB«. pnam=PROCedurname.

\# name
GO# name
ON aexp **GO**# nam,nam,nam
TRAP #name
RESTORE # name

Labeldefinition, das # entspricht dem PROC. In Turbo-Basic XL läßt sich der unstrukturierte Sprungbefehl GOTO wenigstens dadurch lesbarer machen, daß »GOTO zeilennummer« durch »GO# name« ersetzt wird. Auch bei TRAP und RESTORE können Marken verwendet werden. Dabei wird die Kennung »#« verwendet, um dem Interpreter mitzuteilen, daß keine Zeilennummer, sondern ein Name folgt. Übrigens ist »GO# name« schneller als »GOTO zeile«. Die Marken für GO#, TRAP# und RESTORE# werden mit # gekennzeichnet, die für EXEC mit PROC. Diese strikte Trennung dient zur Erhöhung der Gliederung und Transparenz von Programmen.

POP

POP gilt für Unterprogramme mit GOSUB und EXEC ebenso wie für die Schleifen »FOR ... NEXT«, »REPEAT ... UNTIL«, »WHILE ... ENDWHILE« und »LOOP ... ENDLOOP«. Auf dem Runtimestack belegen GOSUB, EXEC, REPEAT, WHILE und LOOP jeweils 4 Byte (wie GOSUB in Atari-Basic). »FOR ... NEXT« belegt jedoch 13 statt 12 Byte, da jetzt 256 verschiedene Variablen zur Verfügung stehen. Auf dem Stack werden jetzt aber nicht mehr die Zeilennummern gespeichert, ein Grund für das langsame Atari-Basic, sondern die Speicheradressen dieser Zeilen. Dadurch werden Schleifen schneller. Außerdem sinkt die Geschwindigkeit der Schleifenwiederholung nicht mit steigender Entfernung vom Programmanfang! Trotzdem konnte einer der Vorzüge von Atari-Basic beibehalten werden: Ein Programm, das beispielsweise durch einen Programmfehler angehalten wurde, kann man editieren und dann mit CONT oder GOTO fortsetzen. Dabei werden weder die Variablen noch der Stack gelöscht. So lassen sich auch Programme schreiben, die ihre DATA-Zeilen selbst generieren oder nicht mehr benötigte Programmteile löschen (mit POKE 842,13, danach verhält sich der Computer so, als würde ständig RETURN gedrückt, bis POKE 842,12 den Normalzustand wieder herstellt). Zum Löschen von Programmteilen gibt es in Turbo-Basic XL den DEL-Befehl.

- -

Dies ist eine spezielle REM-Anweisung. Bei der Eingabe einer Programmzeile wird alles nach »- -« nicht beachtet. Bei LIST werden aber nicht 2, sondern 30 Minuszeichen gedruckt. Diese Anweisung belegt sogar ein Byte weniger Speicherplatz als ein REM ohne Text. Eine entsprechende REM-Zeile im normalen Basic benötigt wesentlich mehr Speicherplatz.

LIST

Beim LISTen von Programmen werden Schleifen optisch durch Einrücken um jeweils zwei Leerzeichen hervorgehoben. Programme sind so übersichtlicher und strukturierter. Außerdem lassen sich auf diese Art und Weise manche Fehler vermeiden. Bei unsauberer Programmierung (mehrere NEXT zu einem FOR, mehrere ENDPROCs zu einem PROC...) gerät das Listing, ebenso wie der Interpreter, allerdings in Unordnung. Solche Konstruktionen, die in Atari-Basic und vielen anderen Basic-Versionen nur schwer zu umschreiben sind, können in Turbo-Basic XL leicht durch IF-ELSE-ENDIF oder EXIT ersetzt werden. Daraus resultieren wieder leichter lesbare Programme.

★L -

schaltet die Tabulierung ab. Dies kann notwendig sein, um lange Programmzeilen zu editieren oder um Platz beim Speichern auf Diskette (LIST"D:X") zu sparen. Die Trennzeile »- -« wird nach »★L -« nur als doppeltes statt dreißigfaches Minus-Zeichen gelistet.

★L
★L +

schaltet die Tabulierung wieder ein (Normalzustand nach dem Laden des Interpreters). Übrigens gibt es jetzt auch ein LIST von einer Programmzeile bis zum Programmende. »LIST 3000,« listet ab Zeile 3000 bis zum Programmende, »LIST"P:",3000,« entsprechend auf den Drucker (das einzelne Komma »,« nach der Zeilennummer veranlaßt den Computer, automatisch 32767 für die höchstmögliche Zeilennummer zu ergänzen).

Neue Fehlermeldungen

Es gibt folgende neue Fehlermeldungen:

ERROR - 22 ?NEST

Schachtelungsfehler, tritt auf, wenn das zu einem WHILE gehörende ENDWHILE nicht gefunden wird, oder das ENDIF zu einem IF, oder auch, nach »★F +«, das NEXT zu einem FOR. Beim Verlassen von Unterprogrammen (durch RETURN oder ENDPROC) werden Schleifen abgebrochen, dies gilt sowohl – wie gewohnt – für die »FOR ... NEXT«-Schleife, wie auch für die anderen unter Turbo-Basic XL zur Verfügung stehenden Schleifen.

ERROR - 16 ?GOSUB

Zu einem GOSUB fehlt RETURN.

ERROR - 13 ?FOR

Zu einem NEXT fehlt FOR.

ERROR - 23 ?WHILE

Zu einem WEND fehlt WHILE.

ERROR - 24 ?REPEAT

Zu einem UNTIL fehlt REPEAT.

ERROR - 25 ?DO
Zu einem LOOP fehlt DO.
ERROR - 28 ?EXEC
Zu einem ENDPROC fehlt EXEC.
ERROR - 29 ?PROC
Eine unbekannte Prozedur wurde aufgerufen.
ERROR - 30 ?#
Eine unbekannte Marke wurde verwendet.
ERROR - 27 XPROC
(Executing PROC). Diese Fehlermeldung tritt auf, wenn eine PROC-Anweisung ausgeführt wird. Prozeduren dürfen nur von EXEC aufgerufen werden.
ERROR - 26 ?EXIT
EXIT ohne Schleife.
ERROR - 15 ?DEL
Das GOSUB zu einem RETURN, NEXT zu einem FOR, REPEAT zu einem UNTIL ... wurde gelöscht. In Atari- und in Turbo-Basic XL lassen sich Programme editieren, ohne Variablenwerte oder den Stapel zu zerstören. Wenn dann bei Rückkehr aus einem Unterprogramm (Schleife) die entsprechende Zeile gelöscht oder verändert wurde, kann dieser Fehler passieren. Dies tritt auch auf, wenn ein in ein Programm eingebautes DEL sich selbst löscht. Alle Fehlernummern, die Atari-Basic ohne irgendwelche Texte ausgibt, werden in Turbo-Basic XL grundsätzlich mit einem kurzen Text ergänzt (beispielsweise »138 TIMEOUT«, »29 ?PROC« etc.). Ausführlichere Erläuterungen können dem Atari-Basic Referenz Manual oder der DOS-Anleitung entnommen werden. Längere Texte würden, bei den insgesamt 60 zur Verfügung stehenden Fehlern, den Platzbedarf des Interpreters noch wesentlich erhöhen.
DEL von,bis
Löscht die Programmzeilen von - bis (jeweils einschließlich).
RENUM alt,neu,incr
Numeriert alle Programmzeilen ab Zeile alt um. Die neuen Zeilennummern beginnen bei neu und werden jeweils um incr erhöht. Alle Zeilennummern vor alt bleiben unverändert. Dieser Befehl ändert auch die Zeilennummern nach GOTO, GOSUB, TRAP, RESTORE, LIST, DEL, ON-GOTO und ON-GOSUB. Bei Verwendung undefinierter Zeilennummern wird die entsprechende negative Zahl eingesetzt (beispielsweise GOTO-100). Bei berechneten Sprüngen (GOTO VAR, GOSUB 100+10★A, RESTORE A★10+1000) versagt RENUM. Wenn hinter dem Befehl eine Zahl folgt »GOTO 1000+A★10«, wird die Zahl wie bei einem normalen Befehl »GOTO 1000« behandelt. Der nachfolgende Programmteil der Zeile bleibt jedoch unverändert. Wenn keine Zahl, sondern ein Variablenname oder eine Klammer folgt, wird der Befehl überhaupt nicht verändert.

DUMP
DUMP filespec
Dieser Befehl erzeugt eine Liste der verwendeten Variablen. Wie bei LIST kann die Ausgabe auch auf einem Drucker erfolgen (DUMP "P:").
Ein Beispiel:

A = 100	numerische Variable
B(10,1	Array, DIM B(9) oder DIM B(9,0)
C(0,0	undimensioniertes Array. Bei Arrays werden die beiden möglichen Dimensionen stets um 1 erhöht angezeigt.
D(10,10	DIM D(9,9)
E$ 10,20	String, LEN=10, DIM E$(20)
F$ 0,0	nicht dimensionierter String
G$ 0,10	DIM G$(10), LEN(G$)=0
H PROC 100	PROC H in Zeile 100
I # 120	Marke I in Zeile 120
J ?	Undefinierte Marke oder PROC

Die Ausgabe der Variablen/Marken erfolgt in der Reihenfolge, wie sie in der Variablentabelle gespeichert sind.
TRACE
TRACE +
Schaltet den TRACE-Modus ein. Das heißt, die Nummer jeder ausgeführten Zeile wird in eckigen Klammern auf dem Bildschirm ausgedruckt.
TRACE -
Hebt TRACE wieder auf. Der TRACE-Modus wird außerdem aufgehoben, sobald eine Fehlermeldung auftritt. Die Zeilennummern eines PROC oder # werden bei Aufruf mit EXEC oder GO# nicht ausgegeben.
★B
★B +
Nach diesem Befehl wird das Drücken der <BREAK>-Taste wie jeder andere Fehler behandelt. Eine Programmunterbrechung läßt sich mit TRAP abfangen und so vor einer versehentlichen Unterbrechung schützen.
★B -
Hebt den oben erwähnten Modus wieder auf. Bei RUN wird »★B -« automatisch ausgeführt.

Befehle

Erläuterung:
< = > steht für: entspricht im normalen Basic
DPOKE adr,word
Doppel-Byte-POKE < = > POKE adr,word-256★INT(wort/256):POKE adr+1,INT(wort/256)
MOVE source,dest,count
Blocktransfer < = > FOR I=0 TO COUNT-1:POKE dest+I,PEEK(source+I):NEXT I
Mit »MOVE 57344,NEUCHARSET,1024« läßt sich beispielsweise der Zeichensatz kopieren.
-MOVE source,dest,count
Blocktransfer, zuerst wird das letzte Byte verschoben. Dies ermöglicht, einen Speicherbereich zu einer höheren Adresse zu verschieben, ohne daß es bei Überlappungen (wenn source+count>dest) zu Zerstörungen kommt. < = > FOR I=COUNT-1 TO 0 STEP -1:POKE dest+I,PEEK(source+I):NEXT I. MOVE kann auch benutzt werden, um einen Speicherbereich zu füllen. Beispiel: »POKE DPEEK(88),128:MOVE DPEEK(88),DPEEK(88)+1,959«. Dies schreibt den Textbildschirm mit dem Bildschirmcode für das inverse Leerzeichen voll. Obwohl hier viele, eigentlich unnötige Ladevorgänge stattfinden, ist dies viel schneller als eine Basic-Schleife.
BPUT #n,adr,len
Blockschreiben < = > FOR I=0 TO len-1:PUT #n,PEEK(adr+I):NEXT I
BGET #n,adr,len
Blocklesen < = > FOR I=0 TO len-1:GET #n,A:POKE adr+I,A:NEXT I
Mit diesen Befehlen kann man Speicherbereiche mit maximaler Geschwindigkeit speichern und laden. Beispiel:
OPEN #1,8,0,"D:BILD.PIC":BPUT #1,DPEEK(88),7680:CLOSE #1
OPEN #1,4,0,"D:BILD.PIC":BGET #1,DPEEK(88),7680:CLOSE #1
Speichern oder Laden eines Grafik-8-(oder 9,10,11,15)-Bildes auf Diskette. Achtung: Die Zahl 7680 muß je nach Grafikmodus geändert werden (sonst werden nicht mehr zum Bildspeicher gehörende Speicherbereiche mitgelesen oder -geschrieben)
%PUT
Mit diesem Befehl lassen sich Zahlen schneller und kompakter auf Diskette oder in einer RAM-Disk speichern. Dabei werden immer sechs Byte gepeichert.

%GET
Zum Lesen von Zahlen, die mit %PUT auf ein Speichermedium gespeichert wurden. Ein Beispiel für die Anwendung von %PUT und %GET finden Sie auf Seite 70.
FILLTO x,y
Kurzschreibweise, schneller und übersichtlicher als: »POSITION x,y:XIO 18,#6,0,0,"S:"FCOLOR n«
Wählen der Farbe für FILLTO. In Standard-Basic heißt dieser Befehl »POKE 765,n«.
CLS
CLS #6
Bildschirmlöschen. CLS <=> A=PEEK(766):POKE 766,0:POSITION 0,0:? (#6;)CHR$(125);:POKE 766,A
PUT n
<=> ?CHR$(n); beispielsweise »PUT 253« für »? CHR$ (253);«. Bei PUT, GET, INPUT... ist auch die Angabe »#0« möglich. Die Benutzung von »IOCB 0« (mit »CLOSE« oder »OPEN #0«) verhindert allerdings das ordnungsgemäße Arbeiten des Bildschirm-Editors, gegebenenfalls <SYSTEM-RESET> drücken. Fehlt bei PUT die #-Angabe, so wird automatisch »IOCB #0« benutzt.
GET KEY
<=> OPEN #7,4,0,"K:":GET #7,KEY:CLOSE #7. Wartet auf Tastendruck und weist der Variablen KEY den ATASCII-Wert der gedrückten Taste zu (der Variablenname ist frei wählbar).
DIM
Beim DIM-Befehl werden Arrays und Strings automatisch gelöscht, also auf 0 gesetzt. DIM A(100) <=> DIM A(100):FOR I=0 TO 100:A(I)=0:NEXT I
INPUT "text",var,var...
INPUT "text";var,var...
INPUT ähnelt jetzt sehr stark dem entsprechenden INPUT-Befehl unter Microsoft-Basic. Ein Text nach INPUT erspart die sonst nötigen PRINT-Befehle. Folgt nach dem Text ein Semikolon statt eines Kommas, so wird zusätzlich ein ? ausgegeben. Mit »INPUT " ";A« erreicht man ein Input ohne das manchmal störende »?«.
TEXT x,y,sexp
TEXT x,y,exp
Schreibt einen Text in ein Grafik-Bild. »x,y« stellt die Position der oberen linken Ecke des ersten Zeichens des String-Ausdrucks (wird in Bildpunkten gezählt) dar. Beispiel: »GRAPHICS 8:TEXT 50,90,"Turbo-Basic":TEXT 70,95,1000«. Im Gegensatz zum normalen PRINT-Befehl darf nach TEXT nur ein Ausdruck erfolgen (keine Liste mit Komma oder Semikolon). Außerdem werden Texte am Zeilenende abgebrochen, es gibt auch kein Scrolling.
CIRCLE x0,y0,r
CIRCLE x0,y0,xr,yr
Zeichnet einen Kreis um den Punkt x0,y0 mit dem Radius r. Bei Angabe zweier (unterschiedlicher) Radien für die x- und y-Richtung entstehen Ellipsen.
PAINT x,y
Füllt eine geschlossene Figur, beispielsweise einen Kreis, mit der mit COLOR gewählten Farbe. Dieser Befehl kann fast jede beliebige Figur mit einer bestimmten Farbe füllen. Da hierbei eine für den 6502 angepaßte, rekursive Funktion benutzt wird, ist der Komplexität der auszufüllenden Figur durch den freien Speicherplatz eine Grenze gesetzt. Im Extremfall würde ein Arbeitsbereich von etwa 90 KByte RAM benötigt. Auch einfache und kleinere Figuren belegen vorübergehend einige hundert Byte. Sollte der Speicher überlaufen (FRE(0) zu klein), so gibt es die Fehlermeldung »ERROR - 2 MEM«. Die PAINT- und die TEXT-Routine benutzen eigene, schnelle Plot-Routinen, da hierbei viele nebeneinander liegende Punkte relativ einfach berechnet werden können. Hiervon machen die Routinen im Atari-Betriebssystem leider keinen Gebrauch.

TIME$= siehe unten
PAUSE n
Unterbricht die Programmausführung für n/50-Sekunden. PAUSE ersetzt das ungenaue und speicherplatzfressende Timing mit leeren FOR-NEXT-Schleifen. In Atari-Basic wird oft die Potenzierung (A=1^1) für eine kleine Verzögerung benutzt. Da Turbo-Basic XL bei der Potenzierung besonders schnell ist, sollte dafür beispielsweise »PAUSE 9« benutzt werden.
DSOUND voice,freq,dis,vol
Ähnlich dem normalen SOUND-Befehl. Der Atari-Computer kann zwei normale Stimmen zu einer zusammenfassen. Die Frequenzauflösung beträgt dann 16 Bit (0..65535) statt 8 Bit (0..255). Die resultierende Frequenz berechnet sich (in Hertz) zu: 1789790/(2★freq+14) statt sonst 63921/(2★freq+2). Diese Werte stammen aus einer amerikanischen Veröffentlichung und unterscheiden sich eventuell etwas von den Werten bei den deutschen Atari-Versionen, da sie von der Videofrequenz abgeleitet sind, die in den USA einen anderen Wert besitzt (siehe auch Seite 124).
DSOUND
Kurzform für: FOR I=0 TO 3:SOUND I,0,0,0:NEXT I
CLOSE
Kurzform für: FOR I=1 TO 7:CLOSE #I:NEXT I

Funktionen

DPEEK(adr)
Doppel-Byte-PEEK <=> PEEK(adr)+256★PEEK (adr+1)
INKEY$
Spezialvariable. Wenn eine Taste gedrückt wird, enthält INKEY$ das entsprechende Zeichen. Wenn nicht, enthält sie einen Leerstring (""). So läßt sich ein Tastendruck verarbeiten, ohne den Programmablauf zu unterbrechen.
INSTR(A$,B$)
INSTR(A$,B$,i)
Sucht einen String B$ in einem (längeren) String A$. Wenn gefunden, wird die Position von B$ in A$ zurückgeliefert, sonst eine Null. »i« stellt den Index (Position) dar, ab dem die Suche beginnen soll.
UINSTR(A$,B$)
UINSTR(A$,B$,i)
Ähnlich INSTR. Die Bits 7 und 5 der einzelnen Zeichen werden nicht beachtet. Bei der Suche nach »MODEM« läßt sich so auch »Modem«, »MoDeM« oder der entsprechende inverse Text finden (UINSTR steht für UppercaseINSTR). Als Nebeneffekt werden bei der Suche nach Zahlen oder Satzzeichen auch die Spezial-Zeichen gefunden ("!"=CTRL-A, "0"=CTRL-P etc.).
ERR
Kurzform für »PEEK(195)« zur Ermittlung des Fehlercodes.
ERL
Kurzform für »PEEK(186)+256★PEEK(187)« oder »DPEEK (186)« zur Ermittlung der Zeile, in der ein Fehler aufgetreten ist. ERR und ERL sollten in TRAP-Routinen verwendet werden.
TIME
Spezialvariable, enthält die Zeit (vom internen Timer RTCLOCK des Atari-Computers, ausschlaggebend sind die Speicherzellen 18 bis 20) in 1/50-Sekunden.
TIME$
Spezialvariable, enthält die Zeit als sechsstelligen String im Format hhmmss (hh=Stunde 00 bis 23, mm=Minute 00 bis 59, ss=Sekunde 00 bis 59).

TIME$=
Zum Stellen der Uhr. »TIME$= "151520"« stellt die Uhr auf 15 Uhr, 15 Minuten und 20 Sekunden. Die Variable TIME läßt sich nicht direkt verändern. Statt dessen wird entweder »TIME$=« verwendet oder entsprechend in die Speicherstellen 18 bis 20 gePOKEt. Die Uhr geht nicht ganz genau, da die Frequenz, mit der im Atari die Fernsehbilder erzeugt werden, nicht genau 50 Hz beträgt. TIME$ wird von dieser Frequenz abgeleitet.
FRAC(exp)
Diese Funktion ermittelt den Nachkommaanteil einer Zahl. »FRAC(exp)« ist nicht immer gleich »exp-INT(exp)«, da INT die nächstkleinere Zahl ermittelt. So ergibt »? INT(-0.3)« -1, »? FRAC(-0.3)« ergibt -0.3.
TRUNC(exp)
Diese Funktion ermittelt den ganzzahligen Anteil einer Zahl. Dies ist die zu FRAC komplementäre Funktion. »? TRUNC(-0.3)« ergibt 0.
RND
RND (irgendwas) kann in Turbo-Basic XL abgekürzt werden, indem man die Klammern wegläßt. Es wird also RND statt RND(0) verwendet (speicherplatzsparend).
RAND(n)
Dies ist die Kurzschreibweise für »TRUNC(RND(0)★n)« und erzeugt eine ganze Zufallszahl zwischen einschließlich 0 und (ausschließlich) n.
HEX$(exp)
Ähnlich STR$. HEX$ wandelt die Integerzahl exp (0<=exp<=65535) in einen sedezimalen (hexadezimalen) String um. Wenn exp kleiner als 256 ist, ergibt dies einen zweistelligen String, sonst einen vierstelligen.
DEC(sexp)
Ähnlich VAL, Umkehrung zu HEX$. Der String sexp wird in eine dezimale Integerzahl gewandelt. Wenn sexp mehr als vier gültige sedezimale Ziffern enthält, so gelten nur die letzten vier.
$aaaa
Sedezimalzahl im Programm. Beispiel:
FOR I=$0600 TO $067F:READ A:POKE I,A:NEXT I
statt:
FOR I=1536 TO 1663:READ A:POKE I,A:NEXT I
& Binäres AND
! Binäres OR
EXOR Binäres exclusiv OR
Diese drei Operatoren arbeiten mit 16-Bit-Integern. Also mit Zahlen zwischen 0 und 65535 und nicht mit Boolschen Größen (1 oder 0) wie die Operatoren AND, OR und NOT.
DIV
Division ohne Rest: a DIV b <=> TRUNC(a/b)
MOD
Bestimmung des Divisionsrestes (Modulo): a MOD b <=> a-b*TRUNC(a/b)
%0 %1 %2 %3
Die Zahlen 0 bis 3 sind als Konstanten definiert. Die Verwendung einer Zahl (auch $aaaa) im Programm kostet jedesmal 7 Byte, die Verwendung einer Variablen, unabhängig von der Länge des Namens, jedoch nur 1 oder 2 Byte. Auch die Verwendung von %0 bis %3 belegt ebenfalls jeweils 1 Byte, aber keinen Eintrag in der auf 256 Variablen begrenzten Variablentabelle.

In Strings oder in zwischen Anführungszeichen stehendem Text ist es jetzt möglich, »"« (entspricht CHR$(34)) durch doppelte Anführungszeichen ("") einzufügen. Beispiel: »?"TEST""TEXT"« erzeugt den Ausdruck: »TEST"TEXT«. In Atari-Basic muß man dafür schreiben: »?"TEST";CHR$(34);
"TEXT"«. Bei Zuweisungen (A$="TEST""TEXT") ist das (in Atari-Basic) noch etwas umständlicher.

Turbo-Basic XL wandelt bei der Eingabe von Programmzeilen automatisch kleine in große und inverse in normale Buchstaben um. Dies gilt natürlich nicht zwischen Anführungszeichen (") sowie nach REM oder DATA-Befehlen. Somit lassen sich Programme auch mit Kleinbuchstaben eingeben, ohne ständig zwischen Groß- und Kleinbuchstaben (mit der <CAPS>-Taste) hin- und herzuschalten oder ständig <SHIFT> drücken zu müssen.

In Variablen- und Prozedurnamen ist außer den Buchstaben und Ziffern auch das Unterstreichungszeichen »_« (<SHIFT ->) zugelassen. So sind endlich Namen wie MAX_LEN oder PROC SORT_KUNDEN erlaubt.

Diskettenbefehle

DIR
DIR "D1:★.★"
Ausgabe des Inhaltsverzeichnisses einer Diskette auf dem Bildschirm. »DIR"D1:★.★"« listet alle Files von Diskette 1, »D2:A★.★« alle Files von Diskette 2, deren Name mit A beginnt. »D1:★.★« kann entfallen, dann wird automatisch »D:★.★« eingesetzt (ähnlich wie im DOS-Menü mit »A RETURN RETURN«). Bei diesem Befehl sind Stringvariablen wie auch bei den folgenden Befehlen zulässig.
RENAME "D:OLD,NEW"
Benennt das File OLD in NEW um. Entspricht: »XIO 32,#7,0,0,"D2:OLD,NEW"« oder der DOS-Menü Auswahl E.
DELETE "D:FILE"
Löscht ein File (entspricht »XIO 33« oder der DOS-Menü Auswahl D).
LOCK "D:FILE"
Schützt ein File gegen Überschreiben (entspricht »XIO 35« oder der DOS-Menü Auswahl F).
UNLOCK "D:FILE"
Hebt den Schreibschutz wieder auf (entspricht »XIO 36« oder der DOS-Menü Auswahl G).
BLOAD "D:FILE.OBJ"
Lädt ein binäres File (entspricht der DOS-Menü Auswahl L mit »/N« nach dem Filenamen).
BRUN "D:FILE.OBJ"
Lädt ein binäres File und startet es, wenn eine RUN-Adresse im File enthalten ist (entspricht der DOS-Menü Auswahl L, ohne angehängtes »/N«).

Damit sind die zusätzlichen Befehle und Funktionen von Turbo-Basic XL erklärt. Selbstverständlich stehen darüber hinaus noch alle anderen Befehle, die man vom normalen Basic her kennt, zur Verfügung. Mit Turbo-Basic XL wird die Programmierung auf dem Atari 800XL/130XE zur wahren Freude. Befehle, die man sich schon lange gewünscht hat, stehen jetzt endlich zur Verfügung.
Autostart
Mit Turbo-Basic XL ist es möglich, nach dem Einschalten des Computers ein Basic-Programm automatisch zu laden und zu starten. Dazu benennen Sie Ihr Basic-Programm einfach in AUTORUN.BAS um. Vergessen Sie nicht: Das eigentliche Turbo-Basic muß dann mit dem Namen AUTORUN.SYS auf Diskette vorliegen.

(Frank Ostrowski/Werner Breuer/hf)

Steckbrief	
Programm:	Turbo-Basic XL
Sprache:	Assembler
Eingabehilfe:	AMPEL
Datenträger:	Diskette, Kassette

```
0000:FF FF E7 02 E8 02 29 36<88>
0008:00 5E 87 5E 00 18 18 FF<E5>
0010:3C 3C 66 42 00 00 00 06<14>
0018:66 7E 76 66 00 00 18 3C<7C>
0020:66 7E 66 66 00 3E 73 33<28>
0028:3E 30 30 78 00 7C 37 33<64>
0030:3E 30 30 78 00 00 60 63<EC>
0038:36 3C 39 63 00 01 01 0F<D4>
0040:03 03 F6 E4 00 80 80 F0<7C>
0048:C0 C0 60 20 00 01 03 06<16>
0050:06 06 03 01 00 E1 33 06<35>
0058:06 06 33 E1 00 C7 63 33<46>
0060:36 36 66 CE 00 0E 9C 9C<7F>
0068:F6 66 66 67 00 7E 33 33<A3>
0070:3E 30 30 78 00 67 66 66<58>
0078:66 66 66 3C 00 FF 31 31<A4>
0080:31 31 31 7B 00 FB 81 81<A1>
0088:F1 81 81 FB 00 F8 8C 8C<C4>
0090:F8 B0 98 CC 00 21 93 21<33>
0098:A2 00 BD 00 E0 9D 00 5C<52>
00A0:BD 00 E1 9D 00 5D BD 00<54>
00A8:E3 9D 00 5F E8 D0 EB A9<34>
00B0:10 8D C6 02 A9 5C 8D C5<49>
00B8:02 8D F0 02 8D F4 02 A9<E4>
00C0:45 8D 44 03 A9 21 8D 45<67>
00C8:03 A9 4F 8D 48 03 8E 49<11>
00D0:03 A9 0B 8D 42 03 20 56<93>
00D8:E4 8E F0 02 60 7D 9B 7F<35>
00E0:7F 20 00 01 02 03 04 05<61>
00E8:06 07 9B 7F 7F 08 09 0A<31>
00F0:0B 0C 0D 0E 0F 10 9B 9B<86>
00F8:7F 20 20 20 20 54 55 52<50>
0100:42 4F 2D 42 41 53 49 43<D9>
0108:20 58 4C 20 31 2E 35 9B<01>
0110:7F 20 28 63 29 20 31 39<6A>
0118:38 35 20 46 72 61 6E 6B<08>
0120:20 4F 73 74 72 6F 77 73<2A>
0128:6B 69 9B 9B E2 02 E3 02<16>
0130:00 21 80 20 A1 20 A9 80<C3>
0138:A0 20 85 0C 84 0D A9 FF<0C>
0140:8D 01 D3 20 00 00 A9 FE<74>
0148:8D 01 D3 A5 80 A4 81 8D<90>
0150:E7 02 8C E8 02 4C 5D E6<36>
0158:00 22 C9 22 A3 24 A3 24<4D>
0160:0B CB 28 C2 1D F1 B4 C2<93>
0168:00 E4 70 F5 BF F3 8B F4<95>
0170:9D FF 9D FF 9A FF FC F5<D9>
0178:11 25 E0 F5 A0 C8 4A C5<90>
0180:5B F5 57 DC A0 C8 6A F5<FC>
0188:65 E6 CC C3 E1 C2 43 C3<C4>
0190:0C C4 18 C4 2F C4 C8 C3<A3>
0198:04 F6 44 25 F2 DC 5A DC<66>
01A0:75 CA 62 CA 18 F8 32 F5<B2>
01A8:B0 F5 D3 F7 F2 DC 67 C4<5B>
01B0:54 C4 53 C2 74 C2 1B C2<F7>
01B8:FE 24 38 C2 AE C1 95 C4<05>
01C0:D9 C1 97 DD 8A C3 92 C3<49>
01C8:00 E4 DA F8 32 25 00 D8<94>
01D0:0A D8 FF F6 1E F4 9B F7<FA>
01D8:84 F7 B9 F7 8D F5 9A F5<49>
01E0:DA C5 DD C5 35 C2 1E F4<E9>
01E8:BB F8 52 F8 A0 C5 C6 C5<7A>
01F0:C9 C5 C3 C5 CC C5 A2 DC<CF>
01F8:74 FB B3 F8 82 F8 64 F8<FC>
0200:2E C2 92 F3 A3 24 E4 33<AD>
0208:00 C0 40 C0 58 FF 37 FC<49>
0210:74 35 77 35 AF F8 A3 24<7C>
0218:9B F5 D0 FD 3F FF D9 C1<59>
0220:80 DE 18 C6 1B C6 00 23<52>
0228:15 23 EA 20 24 23 4C 2B<2C>
0230:E5 44 3A 2A 2E 2A 9B 53<33>
0238:3A 9B 43 3A 9B 50 3A 9B<31>
0240:24 23 2E 23 AD 01 D3 29<2F>
0248:FC 09 02 8D 01 D3 60 3A<E3>
0250:23 28 36 45 D8 4E D8 65<17>
0258:D8 55 D8 5C D8 6C D8 8F<0D>
0260:D8 26 D8 14 D8 1D D8 2F<A9>
0268:D8 95 D8 84 D8 73 D8 A3<D4>
0270:24 89 C6 4C C6 1C C8 45<59>
0278:D8 4E D8 65 D8 55 D8 5C<6D>
0280:D8 6C D8 A3 24 38 D8 A2<44>
0288:C7 94 C6 90 C6 89 C6 90<76>
0290:C6 87 C6 34 DA 63 DA 59<FE>
0298:25 B8 D9 A6 D9 72 D9 D8<AC>
02A0:D9 55 DB 4C DB 89 D9 43<78>
02A8:DB 84 DA 93 D9 70 DB 5E<49>
02B0:DB 67 DB 79 DB CD D8 B6<5E>
02B8:DA FF DA DE D9 E2 D9 E6<68>
02C0:D9 EA D9 1A 25 69 DC 77<65>
02C8:DC B8 DD 4A DA 85 DC 2A<B3>
02D0:DA C2 D9 E4 DA C1 DA FC<4D>
02D8:FA C4 FA D8 DC A3 24 86<BF>
02E0:DA AD DA E7 DA 9E D8 AC<89>
02E8:D8 A6 D8 A3 D8 A3 24 B5<CE>
02F0:DD E4 DC E9 DC 00 00 00<45>
02F8:00 00 00 00 00 00 00 00<FC>
0300:00 00 20 20 20 20 20 20<ED>
0308:2C 28 22 22 28 1E 1A 1C<9E>
0310:32 04 32 32 30 30 30 30<6A>
0318:30 30 2E 2E 32 32 32 32<9B>
0320:32 04 32 32 32 32 32 32<58>
0328:32 32 32 32 32 32 32 32<2D>
0330:32 32 32 32 32 32 32 32<15>
0338:32 32 32 2A 2A 32 32 2A<75>
0340:32 32 28 32 32 32 28 00<BE>
0348:32 32 32 32 32 32 32 00<1A>
0350:32 32 32 00 00 00 00 00<22>
0358:00 00 00 00 00 00 00 00<5E>
0360:20 20 20 20 20 20 2C 28<64>
0368:22 22 28 1D 1A 1C 02 30<13>
0370:01 01 30 30 30 30 30 30<02>
0378:2D 2D 02 02 02 02 02 03<3F>
0380:02 02 02 02 02 02 02 02<86>
0388:02 02 02 02 02 02 02 02<8E>
0390:02 02 02 02 02 02 02 02<96>
0398:02 2A 2A 02 32 2A 02 02<C7>
03A0:28 02 32 32 28 00 32 02<2A>
03A8:02 32 32 32 32 00 02 32<4D>
03B0:32 EE 01 D3 20 56 E4 CE<88>
03B8:01 D3 C0 00 60 68 AA CE<AF>
03C0:01 D3 68 40 2C 0F D4 10<82>
03C8:03 6C 00 02 48 8A 48 A9<2F>
03D0:24 48 A9 A4 48 BA BD 05<06>
03D8:01 48 D8 48 8A 48 98 48<CF>
03E0:EE 01 D3 8D 0F D4 6C 22<03>
03E8:02 48 A9 24 48 A9 A6 48<B3>
03F0:08 EE 01 D3 6C 16 02 EE<71>
03F8:01 D3 20 E8 24 CE 01 D3<E4>
0400:60 BD 47 03 48 BD 46 03<43>
0408:48 98 A0 5C 60 EE 01 D3<53>
0410:20 00 00 CE 01 D3 60 20<39>
0418:34 C5 EE 01 D3 AD 8E 20<C7>
0420:AC 8F 20 85 0C 84 0D 6C<73>
0428:0A 00 20 34 C5 EE 01 D3<2F>
0430:4C 71 E4 20 16 E5 EE 01<5A>
0438:D3 A0 01 B1 D4 AA 88 B1<D4>
0440:D4 CE 01 D3 85 D4 86 D5<50>
0448:4C 7D D9 20 0C E5 EE 01<83>
0450:D3 A0 00 91 9B A5 D5 C8<D4>
0458:D0 0B 4C 28 E5 20 0C E5<BB>
0460:D0 F8 EE 01 D3 91 9B AD<D7>
0468:01 D3 29 FC 09 02 8D 01<F6>
0470:D3 60 20 62 25 20 DC 2B<A0>
0478:4C 3A E5 A9 23 48 48 A5<DC>
0480:B0 85 C6 20 16 E5 C6 C6<72>
0488:30 09 A5 D4 48 A5 D5 48<A3>
0490:4C 6A 25 EE 01 D3 A5 B0<ED>
0498:48 6C D4 00 4C 7B F6 A9<3F>
04A0:00 84 A4 85 A5 18 A5 90<C5>
04A8:65 A4 A8 A5 91 65 A5 CD<CB>
04B0:E6 02 90 09 D0 E6 CC E5<61>
04B8:02 90 02 D0 DF 38 A5 90<D4>
04C0:F5 00 85 A2 A5 91 F5 01<70>
04C8:85 A3 18 B5 00 85 97 85<10>
04D0:99 65 A4 85 9B B5 01 85<7D>
04D8:98 85 9A 65 A5 85 9C B5<80>
04E0:00 65 A4 95 00 B5 01 65<29>
04E8:A5 95 01 E8 E8 E0 92 90<BB>
04F0:EE 85 0F A5 90 85 0E EE<E9>
04F8:01 D3 A6 A3 18 8A 65 9A<A2>
0500:85 9A 18 8A 65 9C 85 9C<57>
0508:E8 A4 A2 F0 25 88 B1 99<A4>
0510:91 9B 98 D0 F8 F0 1B C6<15>
0518:9A C6 9C 88 B1 99 91 9B<A1>
0520:88 B1 99 91 9B 88 B1 99<A4>
0528:91 9B 88 B1 99 91 9B 98<A2>
0530:D0 E9 CA D0 E2 CE 01 D3<AD>
0538:60 A9 00 84 A4 85 A5 38<87>
0540:A5 90 F5 00 85 A2 A5 91<C4>
0548:F5 01 85 A3 38 B5 00 85<F9>
0550:99 E5 A4 85 9B B5 01 85<DB>
0558:9A E5 A5 85 9C 38 B5 00<8C>
0560:E5 A4 95 00 B5 01 E5 A5<9A>
0568:95 01 E8 E8 E0 92 90 ED<D1>
0570:85 0F A5 90 85 0E EE 01<37>
0578:D3 A0 00 A6 A3 F0 1D B1<7E>
0580:99 91 9B C8 B1 99 91 9B<33>
0588:C8 B1 99 91 9B C8 B1 99<61>
0590:91 9B C8 D0 EA E6 9A E6<08>
0598:9C CA D0 E3 A6 A2 F0 08<4B>
05A0:B1 99 91 9B C8 CA D0 F8<42>
05A8:CE 01 D3 60 18 4C 9C 2B<8C>
05B0:18 60 38 60 20 9C 2E A5<40>
05B8:D4 F0 F5 A5 E0 F0 ED 45<6F>
05C0:D4 29 80 85 EE A5 E0 29<21>
05C8:7F 85 E0 A5 D4 29 7F 38<33>
05D0:E9 40 38 65 E0 30 DB 05<13>
05D8:EE A8 20 9E 2A 85 DA 85<3D>
05E0:DB 85 DC 85 DD 85 DE 85<CB>
05E8:DF 84 D4 A0 07 46 E5 90<7A>
05F0:30 18 A5 DE 79 F2 22 85<67>
05F8:DE A5 DD 79 EA 22 85 DD<8B>
0600:A5 DC 79 E2 22 85 DC A5<88>
0608:DB 79 DA 22 85 DB A5 DA<F0>
0610:79 D2 22 85 DA A5 D9 79<7A>
0618:CA 22 85 D9 88 10 CE 30<05>
0620:05 F0 03 88 10 C7 A0 07<69>
0628:46 E4 90 30 18 A5 DD 79<CB>
0630:F2 22 85 DD A5 DC 79 EA<8D>
0638:22 85 DC A5 DB 79 E2 22<2B>
0640:85 DB A5 DA 79 DA 22 85<8C>
0648:DA A5 D9 79 D2 22 85 D9<54>
0650:A5 D8 79 CA 22 85 D8 88<CF>
0658:10 CE 30 05 F0 03 88 10<C0>
0660:C7 A0 07 46 E3 90 30 18<0E>
0668:A5 DC 79 F2 22 85 DC A5<B0>
0670:DB 79 EA 22 85 DB A5 DA<1A>
0678:79 E2 22 85 DA A5 D9 79<16>
0680:DA 22 85 D9 A5 D8 79 D2<D1>
0688:22 85 D8 A5 D7 79 CA 22<6A>
0690:85 D7 88 10 CE 30 05 F0<30>
0698:03 88 10 C7 A0 07 46 E2<FC>
06A0:90 30 18 A5 DB 79 F2 22<E0>
06A8:85 DB A5 DA 79 EA 22 85<9F>
06B0:DA A5 D9 79 E2 22 85 D9<EF>
06B8:A5 D8 79 DA 22 85 D8 A5<51>
06C0:D7 79 D2 22 85 D7 A5 D6<5B>
06C8:79 CA 22 85 D6 88 10 CE<D3>
06D0:30 05 F0 03 88 10 C7 A0<0E>
06D8:07 46 E1 90 30 18 A5 DA<1C>
06E0:79 F2 22 85 DA A5 D9 79<C2>
06E8:EA 22 85 D9 A5 D8 79 E2<F0>
06F0:22 85 D8 A5 D7 79 DA 22<A2>
06F8:85 D7 A5 D6 79 D2 22 85<91>
0700:D6 A5 D5 79 CA 22 85 D5<99>
0708:88 10 CE 30 05 F0 03 88<91>
0710:10 C7 4C B4 2D 18 60 38<4F>
0718:60 A5 E0 F0 FA A5 D4 F0<B4>
0720:F4 45 E0 29 80 85 EE A5<A4>
0728:E0 29 7F 85 E0 A5 D4 29<04>
0730:7F 38 E5 E0 18 69 40 30<79>
0738:DE 05 EE A8 20 9E 2A 85<0F>
0740:E6 85 E7 85 E8 85 E9 85<90>
0748:EA 85 E0 85 DA 84 D4 A0<1D>
0750:00 A5 E0 D9 CA 22 D0 28<B7>
0758:A5 E1 D9 D2 22 D0 21 A5<60>
0760:E2 D9 DA 22 D0 1A A5 E3<06>
0768:D9 E2 22 D0 13 A5 E4 D9<6C>
0770:EA 22 D0 0C A5 E5 D9 F2<13>
0778:22 D0 05 A2 00 4C 95 2A<CA>
0780:90 2A A5 E5 F9 F2 22 85<BF>
0788:E5 A5 E4 F9 EA 22 85 E4<45>
0790:A5 E3 F9 E2 22 85 E3 A5<F1>
0798:E2 F9 DA 22 85 E2 A5 E1<11>
07A0:F9 D2 22 85 E1 A5 E0 F9<6E>
07A8:CA 22 85 E0 26 D5 C8 C0<AA>
07B0:08 D0 9E A0 00 A5 E1 D9<D5>
07B8:CA 22 D0 28 A5 E2 D9 D2<D9>
07C0:22 D0 21 A5 E3 D9 DA 22<92>
07C8:D0 1A A5 E4 D9 E2 22 D0<5A>
07D0:13 A5 E5 D9 EA 22 D0 0C<3F>
07D8:A5 E6 D9 F2 22 D0 05 A2<E9>
07E0:01 4C 95 2A 90 2A A5 E6<EC>
07E8:F9 F2 22 85 E6 A5 E5 F9<11>
07F0:EA 22 85 E5 A5 E4 F9 E2<E4>
07F8:22 85 E4 A5 E3 F9 DA 22<21>
0800:85 E3 A5 E2 F9 D2 22 85<26>
0808:E2 A5 E1 F9 CA 22 85 E1<E0>
0810:26 D6 C8 C0 08 D0 9E A0<08>
0818:00 A5 E2 D9 CA 22 D0 28<B9>
0820:A5 E3 D9 D2 22 D0 21 A5<EB>
0828:E4 D9 DA 22 D0 1A A5 E5<F3>
0830:D9 E2 22 D0 13 A5 E6 D9<36>
0838:EA 22 D0 0C A5 E7 D9 F2<BC>
0840:22 D0 05 A2 02 4C 95 2A<A4>
0848:90 2A A5 E7 F9 F2 22 85<26>
0850:E7 A5 E6 F9 EA 22 85 E6<50>
0858:A5 E5 F9 E2 22 85 E5 A5<C3>
0860:E4 F9 DA 22 85 E4 A5 E3<FE>
0868:F9 D2 22 85 E3 A5 E2 F9<60>
0870:CA 22 85 E2 26 D7 C8 C0<19>
0878:08 D0 9E A0 00 A5 E3 D9<8F>
0880:CA 22 D0 28 A5 E4 D9 D2<37>
0888:22 D0 21 A5 E5 D9 DA 22<5C>
0890:D0 1A A5 E6 D9 E2 22 D0<64>
0898:13 A5 E7 D9 EA 22 D0 0C<49>
08A0:A5 E8 D9 F2 22 D0 05 A2<6F>
08A8:03 4C 95 2A 90 2A A5 E8<B0>
08B0:F9 F2 22 85 E8 A5 E7 F9<E0>
```